

A

# MICHAELSON & WALLACE

A Partnership of Intellectual Property and Technology Lawyers

Doc. No. MS-68(116627.1)

June 18, 1999

**PATENT APPLICATION**

1525 U.S. PTO  
09/335640  
06/18/99

Peter L. Michaelson  
NJ, NY, PA Bars

06/01/90  
M. Wallace

Pokotylo

P. Straub  
NJ, NY Bars

Glenn B. Foster  
NY, MA, D.C. Bars

Christopher R. Balzan  
CA Bar

Craig S. Fischer  
CA Bar

Eric. J. Agaard  
CA Bar

Jeremiah G. Murray  
Patent Agent

Counsel:

John T. Peoples  
NY Bar

Ronald L. Drumheller  
NY Bar

Edward M. Fink  
NY, D.C. Bars

Internet Address:  
mail@mandw.com

Web Site:  
http://www.mandw.com

Please reply:

- ☒ Parkway 109 Office Ctr.  
328 Newman Springs Rd.  
P. O. Box 8489  
Red Bank, NJ 07701  
Tel: 732-530-6671  
Fax: 732-530-6584 or  
732-530-6585  
Videoconference:  
732-224-0132 (ISDN)

- ☐ 1001 Partridge Drive  
Suite 330  
Ventura, CA 93003  
Tel: 805-644-4035  
Fax: 805-644-8046

Assistant Commissioner of Patents  
Box **Patent Application**  
Washington, D. C. 20231

Sir:

Enclosed herewith for filing is the following  
**utility patent application:**

Applicants: **Matt CONWAY, Stephen JACQUOT, Dennis PROFFITT,  
George ROBERTSON**

Title of application: **METHODS, APPARATUS AND DATA STRUCTURES  
FOR PROVIDING A USER INTERFACE TO  
OBJECTS, THE USER INTERFACE EXPLOITING  
SPATIAL MEMORY AND VISUALLY INDICATING  
AT LEAST ONE OBJECT PARAMETER**

Total pages of specification: 97 (including 17 pages of  
claims [1-60] and 1 page  
of abstract)

Sheets of drawing: 25 (FIGs. 1-22)

Docket No.: MS-68(116627.1)

**PATENT APPLICATION FILING FEE CALCULATION**

	<u>No. Filed</u>	<u>Less</u>	<u>Rate/Claim</u>	<u>Fee</u>
Total				
Claims	<u>60</u>	-20	40 x \$18.00	\$ 720.00
Independent				
Claims	<u>2</u>	-3	0 x \$78.00	\$ 00.00
Minimum Filing Fee				\$ 760.00
Multiple Dependency Fee (if applicable - \$260.00)				\$ 00.00

-1-

\*\*\*EXPRESS MAIL CERTIFICATE APPEARS ON LAST PAGE\*\*\*

50% Reduction for Small Entity  
(Independent Inventor, Non-profit  
Corporation, or Small Business  
Concern) - appropriate

verified statement attached \$- 00.00

TOTAL FILING FEE \$ 1,480.00

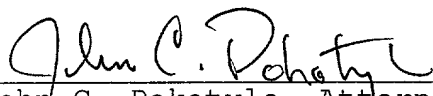
**TOTAL FILING FEE ENCLOSED \$ 00.00**

Also enclosed herewith for filing in connection  
with the enclosed application are:

XX Postcard Receipt (2);  
Check;  
XX Utility Patent Application Transmittal;  
A Recordation Form Cover Sheet and an Assignment of  
the application to: \_\_\_\_\_;  
Oath;  
Declaration and Power of Attorney;  
Disclosure Statement;  
Letter referencing previously filed disclosure  
document; number \_\_\_\_\_ filed \_\_\_\_\_;  
Verified Statement Claiming Small Entity Status;  
Claim to priority (as noted below and included  
in declaration):  
\_\_\_\_\_  
Serial Number \_\_\_\_\_ Filing date \_\_\_\_\_  
Submission of Priority Document and a certified  
copy of a \_\_\_\_\_ patent application or  
inventor's certificate, filed \_\_\_\_\_ and  
serial no. \_\_\_\_\_, upon which a claim to  
priority is made; and  
Other: \_\_\_\_\_.

Respectfully submitted,

MICHAELSON & WALLACE

  
John C. Pokotylo, Attorney  
Reg. No. 36,242  
Customer No. 007265

\*\*\*EXPRESS MAIL CERTIFICATION\*\*\*

"Express Mail" mailing label number: EL344854598US

Date of deposit: June 18, 1999

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents, **Box Patent Application**, Washington, D.C. 20231.



\_\_\_\_\_  
Signature of person making certification

\_\_\_\_\_  
John C. Pokotylo

\_\_\_\_\_  
Name of person making certification

(m68pto/109)

55790-049266

Please type a plus sign (+) inside this box → ☐

PTO/SB/05 (4/98)  
Approved for use through 09/30/2000. OMB 0651-0032  
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

# UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No. EL344854598US

First Inventor or Application Identifier Matt CONWAY

Title Methods, Apparatus and Data Structures  
for Providing a User Interface to ...

Express Mail Label No. EL344854598US

## APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. ☒ \* Fee Transmittal Form ~~PTO/SB/17~~ (3 pgs.)  
(Submit an original and a duplicate for fee processing)
2. ☒ Specification [Total Pages 97]  
(preferred arrangement set forth below)
  - Descriptive title of the invention
  - Cross References to Related Applications,
  - Statement Regarding Fed sponsored R & D
  - Reference to Microfiche Appendix
  - Background of the invention
  - Brief Summary of the invention
  - Brief Description of the Drawings (if filed)
  - Detailed Description
  - Claim(s)
  - Abstract of the Disclosure
3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 25]
4. Oath or Declaration [Total Pages ]
  - a. ☐ Newly executed (original or copy)
  - b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))  
(for continuation/divisional with Box 16 completed)
    - i. ☐ DELETION OF INVENTOR(S)  
Signed statement attached deleting  
inventor(s) named in the prior application,  
see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

\* NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY  
FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT  
IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28)

ADDRESS TO: Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

5. ☐ Microfiche Computer Program (Appendix)
6. Nucleotide and/or Amino Acid Sequence Submission  
(if applicable, all necessary)
  - a. ☐ Computer Readable Copy
  - b. ☐ Paper Copy (identical to computer copy)
  - c. ☐ Statement verifying identity of above copies

## ACCOMPANYING APPLICATION PARTS

7. ☐ Assignment Papers (cover sheet & document(s))
8. ☐ 37 C.F.R. § 3.73(b) Statement ☐ Power of  
(when there is an assignee) ☐ Attorney
9. ☐ English Translation Document (if applicable)
10. ☐ Information Disclosure ☐ Copies of IDS  
Statement (IDS)/PTO-1449 ☐ Citations
11. ☐ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503) (2 ea.)  
(Should be specifically itemized)
13. ☐ \* Small Entity ☐ Statement filed in prior application,  
Statement(s) ☐ Status still proper and desired  
(PTO/SB/09-12)
14. ☐ Certified Copy of Priority Document(s)  
(if foreign priority is claimed)
15. ☐ Other: \_\_\_\_\_

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: \_\_\_\_\_

Prior application information: Examiner \_\_\_\_\_ Group / Art Unit: \_\_\_\_\_

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

## 17. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label

007265

or ☐ Correspondence address below

(Insert Customer No. or Attach bar code label here)

Name

Address

City

State

Zip Code

Country

Telephone

Fax

Name (Print/Type)

John C. Pokotylo, Esq.

Registration No. (Attorney/Agent)

36,242

Signature

*John C. Pokotylo*

Date

6/18/99

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

-1-

**METHODS, APPARATUS AND DATA STRUCTURES FOR PROVIDING A  
USER INTERFACE TO OBJECTS, THE USER INTERFACE EXPLOITING  
SPATIAL MEMORY AND VISUALLY INDICATING AT LEAST ONE  
OBJECT PARAMETER**

**§ 1. BACKGROUND OF THE INVENTION**

**§ 1.1 Field of the Invention**

5           The present invention concerns a user interface  
to objects, such as informational objects for example,  
which are stored on, or accessed via, a machine such as a  
computer for example. The present invention exploits the  
spatial memory of people and maps one or more parameters  
10 or properties of an object to a coordinate of a location  
of an object.

**§ 1.2 Related Art**

15           A brief history of user interfaces is provided  
below. First, known user interfaces are introduced in  
§ 1.2.1. Then, user interfaces which facilitate  
information access, rather than information processing,  
are introduced in § 1.2.2. Finally, needs of users which  
20 have not yet been met by user interfaces are listed in  
§ 1.2.3.

### **§ 1.2.1 KNOWN USER INTERFACES**

The way in which people interact with computing machines has evolved over the last 50 or so years.

5 Initially, these machines were typically used as information processors, and in particular, for performing mathematical operations on numbers. People interacted with such early computing machines by punching and ordering cards to effect a sequence of commands. In  
10 later computing machines, also typically used to perform mathematical operations on numbers, people interacted with such computing machines by setting switches and viewing light emitting diodes to enter commands. With the advent of the desktop personal computer,  
15 people-machine interaction evolved from the use a keyboard to enter lines of commands, discussed in § 1.2.1.1 below, to the use of a keyboard and mouse to manipulate icon metaphors of the real world, discussed in § 1.2.1.2 below.

#### **§ 1.2.1.1 COMMAND LINES**

20 Early personal computers were also used to perform mathematical operations, from engineering applications to accounting applications (e.g.,  
25 spreadsheets). In addition, such early personal computers were used to enter, store, and manipulate information, such as with word processing applications

for example, and to effectively access stored information, such as with relational database applications for example. People typically interacted with such computers by entering commands, in accordance with fairly rigid syntactical rules, or by entering data via a keyboard and viewing results via a video monitor. Unfortunately, since the syntactical rules for interacting with a personal computer were typically not intuitive, people would have to invest a fairly substantial amount of time to master effective interaction with the computer. Until they mastered the syntactical rules, people would often become frustrated when using computers. Casual users would often conclude that learning to interact with computers would not be worth their time. Thus, computer use was usually limited to professionals (such as accountants, engineers, and scientists for example) who needed the computational power of personal computers, and to hobbyists.

#### **§ 1.2.1.2 GRAPHICAL USER INTERFACES**

The advent of graphical user interfaces (or "GUIs") provided a more intuitive way for people to interact with computers. The casual user no longer needed to learn syntactical rules and enter sequences of commands. As personal computers penetrated business to an ever greater degree, probably due to popular and time saving word processing and spreadsheet applications,

operating systems began to employ user interfaces which used an office metaphor which included documents, folders, filing cabinets, trash cans, telephone directories, etc. These so-called desktop GUIs have served their users well over the past decade or so. However, in addition to using computers for data entry, storage, and manipulation, people are using computers for access to information to an ever increasing degree. This recent trend is discussed in § 1.2.2 below.

#### **§ 1.2.2    MIGRATION FROM INFORMATION PROCESSING AND DESKTOP METAPHOR TO INFORMATION ACCESS**

In recent decades, and in the past five (5) to ten (10) years in particular, computers have become interconnected by networks by an ever increasing extent; initially, via local area networks (or "LANs"), and more recently via LANs, private wide area networks (or "WANs") and the Internet. The proliferation of networks, in conjunction with the increased availability of inexpensive data storage means, has afforded computer users unprecedented access to a wealth of content. Such content may be presented to a user (or "rendered") in the form of text, images, audio, video, etc.

The Internet is one means of inter-networking local area networks and individual computers. The popularity of the Internet has exploded in recent years.



Many feel that this explosive growth was fueled by the ability to link (e.g., via Hyper-text links) resources (e.g., World Wide Web pages) so that users could seamlessly transition from various resources, even when  
5 such resources were stored at geographically remote resource servers. More specifically, the Hyper-text markup language (or "HTML") permits documents to include hyper-text links. These hyper-text links, which are typically rendered in a text file as text in a different  
10 font or color, include network address information to related resources. More specifically, the hyper-text link has an associated uniform resource locator (or "URL") which is an Internet address at which the linked resource is located. When a user activates a hyper-text  
15 link, for example by clicking a mouse when a displayed cursor coincides with the text associated with the hyper-text link, the related resource is accessed, downloaded, and rendered to the user. The related resource may be accessed by the same resource server that  
20 provided the previously rendered resource, or may be accessed by a geographically remote resource server. Such transiting from resource to resource, by activating hyper-text links for example, is commonly referred to as "surfing".

25

Although people continue to use computers to enter information, manipulate information, and store information, in view of the foregoing developments people

are using computers to access information to an ever increasing extent. In a departure from the past, the information people want to access is often not created by them (which would typically reside on the person's desktop computer), or even by a company or group to which that person belongs (which would typically reside on a storage server, accessible via a local area network). Rather, given the world wide breadth of the Internet, the information people want to access is usually created by unrelated third parties (or content providers). Unfortunately, GUIs using a desktop metaphor are not particularly well suited for such information access.

New GUIs should therefore help people find information that they want, or that they might want. Unfortunately, the very vastness of available data can overwhelm a user; desired data can become difficult to find and search heuristics employed to locate desired data often return unwanted data (also referred to as "noise").

Various concepts have been employed to help users locate desired data. In the context of the Internet for example, some services have organized content based on a hierarchy of categories. A user may then navigate through a series of hierarchical menus to find content that may be of interest to them. An example of such a service is the YAHOO™ World Wide Web site on

the Internet. Unfortunately, content, in the form of Internet "web sites" for example, must be organized by the service and users must navigate through a predetermined hierarchy of menus. If a user mistakenly  
5 believes that a category will be of interest or include what they were looking for, but the category turns out to be irrelevant, the user must backtrack through one or more hierarchical levels of categories.

10 Again in the context of the Internet for example, some services provide "search engines" which search databased content or "web sites" pursuant to a user query. In response to a user's query, a rank ordered list, which includes brief descriptions of the  
15 uncovered content, as well as a hypertext links (text, having associated Internet address information, which, when activated, commands a computer to retrieve content from the associated Internet address) to the uncovered content is returned. The rank ordering of the list is  
20 typically based on a match between words appearing in the query and words appearing in the content. Unfortunately, however, present limitations of search heuristics often cause irrelevant content (or "noise") to be returned in response to a query. Again, unfortunately, the very  
25 wealth of available content impairs the efficacy of these search engines since it is difficult to separate irrelevant content from relevant content.

The foregoing means permit users to find content of interest to them, or to deliver content that may be of interest to a user. Unfortunately, neither of the foregoing means is particularly well suited when the user wants to "go back" to (or "relocate") information (or content), to revisit that information or to revisit a familiar content provider to view new information (or content). More specifically, a person typically does not want to go through steps of navigating through a hierarchy of menus, or entering a search query to get to favored content or favored content providers.

#### **§ 1.2.2.1 INFORMATION ACCESS USER INTERFACES AND THEIR LIMITATIONS**

In view of the shift towards using computers to access information, a number of user interfaces have been developed to help people revisit or relocate information or content providers. A sample of these user interfaces, as well as perceived limitations of such user interfaces, is presented below.

##### **§ 1.2.2.1.1 BOOKMARKS AND FAVORITES LISTS**

Some so-called "Internet browser" program services, such as Microsoft's Internet Explorer™ for example, permit people to create lists of favorite Internet locations (as located by a Uniform Resource

Locator or "URL") represented by bookmarks.

Unfortunately, as the number of bookmarks in a list increases, a person's ability to find a desired bookmark becomes more difficult. Although people can arrange and  
5 organize such lists of bookmarks into hierarchies, this requires some effort (mental effort is also referred to as "cognitive load") to organize and maintain the hierarchy. Moreover, the use of hierarchies does not fully exploit the spatial memory (This concept has also  
10 been referred to as "where it is is what it is".) of people. Finally, other than textual information regarding the information, such as a title of a web page for example, the bookmarks do not provide the user with any other information about the information.

15  
**§ 1.2.2.1.2 WEB BOOK / WEB FORAGER**

The article, Card, et al., "The WebBook and Web Forager: An Information Workspace for the World-Wide  
20 Web," Proceedings of CHI '96, pp. 111-117 (April 13-18, 1996) (hereafter referred to as "the WebBook article") also recognized the trend towards the use of computers for information access. Focusing on the use of the Internet, the WebBook article noted, among other things,  
25 that web pages are often hard to find, users get lost and have difficulty relocating pages, and users have difficulty organizing found pages. The article then discussed a WebBook™ which simulates, in three

dimensions, a physical book and includes pages, each of which correspond to a web page. Hypertext links are color coded to indicate whether the referenced web page is within the currently opened book or not. If the web page referenced by the link is in the book, activating the hypertext link will effect an animation of flipping pages to the desired web page. If the web page referenced by the link is not in the book, activating the hypertext link will close the currently opened WebBook and, if the web page is in another WebBook on a simulated bookshelf, will open that WebBook to the selected web page. The user can flip or ruffle through the pages of the WebBook using various input techniques. Portions of interest on a web page may be inspected with panning and zooming operations. The Document Lens feature described in the WebBook article is related to U.S. Patent No. 5,670,984 to Robertson (incorporated herein by reference).

Although the WebBook uses screen space efficiently and uses a familiar metaphor, it has a number of limitations. First, the book metaphor limits the number of web pages that can be seen at one time -- most pages are totally occluded. Second, the book metaphor does not exploit spatial memory. That is, it is believed that user will only remember the ordered pages in a linear (or one-dimensional) manner. Finally, the WebBook

does not convey any information about the properties of the information or content.

The WebBook article also discusses a Web Forager which embeds the WebBook and other objects in a hierarchical three-dimensional workspace. The workspace includes four (4) hierarchical levels; namely a focus place, an immediate memory space, a secondary storage area, and a tertiary storage area. In the focus place, objects are rendered in full size and can be acted upon by the user. The immediate memory space uses the metaphor of a desk top. The secondary storage area uses several tiers in Z space, in which objects can be moved. Finally the tertiary storage area uses a bookshelf metaphor. A book selected from the bookshelf will come up to the focus place and an object then in the focus place will be moved into the secondary memory space.

Although the Web Forager exploits, at least to some extent, spatial memory, particularly in the secondary storage area, the interaction between the focus place, immediate memory place, secondary storage area, and tertiary storage area may not be immediately obvious to new users. Further, if more than about 30 objects are placed in the secondary storage place, objects may become occluded. Finally, the use of tiers in the Z dimension of the secondary storage place limits the exploitation of a user's spatial memory.

### § 1.2.2.1.3 WORKSCAPE

Noting the concept of "what it is is where it  
5 is", the article, Ballay, "Designing Workspace<sup>TM</sup>: An  
Interdisciplinary Experience", Human Factors in Computing  
Systems, CHI '94, pp. 10-15 (April 1994) (hereafter  
referred to as "the Workspace article"), discusses a  
three-dimensional user interface for managing documents.  
10 As discussed at the web page on the Internet at  
<http://www.maya.com/Portfolio/workspace.html> (downloaded  
July 7, 1998, revised 1998), Workspace permits users to  
drag documents in the X-Y plane, and also push and pull  
documents in the Z dimension.

15 Although Workspace exploits spatial memory, it  
is apparently not concerned with the problem of object  
occlusion. Furthermore, it is believed that separate  
user operations for (i) dragging an object in the X-Y  
20 plane, and (ii) pushing and pulling documents in the Z  
dimension, will be cumbersome for users in practice.  
Finally, the location of the object does not provide a  
user with any information about any properties of the  
object.



#### **§ 1.2.2.1.4 MAPA**

The product MAPA, offered by Dynamic Diagrams of Providence, RI, uses a three-dimensional display to show the organization of web pages at a web site. As a web page is selected, its children in the web site hierarchy are displayed. The pages are represented by sheets. As a user's cursor hovers over a page, a pop-up title is provided. Although MAPA may be useful for visualizing large amounts of hierarchical data, it lends itself to a predetermined hierarchies (i.e., web pages of a web site). That is, it is believed that the strength of MAPA is presenting information, not building or updating hierarchies of information.

#### **§ 1.2.2.1.5 MacIntosh and Windows NT® Operating Systems**

Some versions of the MacIntosh and Windows NT® Operating Systems permit users to arrange, using a mouse drag operation for example, icons on a two-dimensional surface rendered on the screen of a video monitor. Drag feedback is provided. The user's arrangement of the icons persists. However, the icons represent applications or files or directories -- they are not miniature images, or thumbnails, of objects. Further, the location of the icon does not relay any information regarding any of its properties, unless the user intentionally arranges the icons in such a manner.

#### **§ 1.2.2.1.6 Windows NT® 5.0**

A file manager or windows explorer feature of  
5 Windows NT® 5.0 may be configured to display document or  
file thumbnails in a selected directory. Each thumbnail  
includes a title which persists (i.e., is continuously  
rendered). A user can rearrange these thumbnails on a  
display screen, for example, by using a mouse drag  
10 operation. However, no drag feedback is provided.  
Furthermore, the user arrangement of the thumbnails does  
not persist. That is, when the selected directory is  
left and then reentered, the thumbnails are arranged in a  
default manner, not based on a previously entered user  
15 arrangement. Further, the location of the thumbnail does  
not relay any information regarding any of its  
properties, unless the user intentionally arranges the  
thumbnails in such a manner.

#### **§ 1.2.2.1.7 InfoGrid**

The article: R. Rao, S. K. Card, H. D.  
Jellinek, J. D. Mackinlay, and G. G. Robertson, "The  
25 Information Grid: A Framework for Information Retrieval  
and Retrieval-Centered Applications," UIST'92, pp. 23-32  
(November 15-18, 1992) (hereafter referred to as "the  
InfoGrid article") discusses a user interface for object

(e.g., document) retrieval. The InfoGrid display layout includes (i) an area for entering search parameters, (ii) a control panel which includes tool buttons, (iii) a area for rendering thumbnails of objects returned from a search, (iv) an area to hold selected objects, (v) an area for depicting a search path, and (vi) an area for depicting a document and its properties. The InfoGrid does not exploit a user's spatial memory and, in fact, teaches away from using location-centric workspace retrieval. (See page 31 of the InfoGrid article.) Instead, the InfoGrid permits a user to specify an initial query by providing values for various fields in a search parameter area, after which objects may be retrieved using an iterative process. (See page 26 of the InfoGrid article.)

#### **§ 1.2.2.1.8 DATA MOUNTAIN**

U.S. Patent Application Serial No. 09/152,491 entitled "Methods, Apparatus, and Data Structures for Providing a User Interface, which Exploits Spatial Memory in Three-Dimensions, to Objects" and filed on September 14, 1998 discloses a user interface in which objects, represented by thumbnails, may be positioned on a simulated three-dimensional surface. However, the position of the thumbnails on the three-dimensional surface does not necessarily convey any information about the object -- the user is free to position the object in

the simulated three-dimensional space. It is to be understood, however, that the user may position object thumbnails in accordance with some purposeful, or unconscious, scheme.

5

### **§ 1.2.3 UNMET NEEDS**

As discussed above, there exists a need for a user interface, and in particular a graphical user interface, to information or content. Such a user interface should exploit spatial memory. For example, the user interface should permit continuous movement in the simulated space. The user interface should also be intuitive to minimize the time needed for a user to become familiar with it. For example, the user interface should not be cumbersome or require too many different types of inputs to manipulate objects. The user interface should convey, at minimal cognitive load to the user, properties or parameters of the information to the user. Finally, the user interface should provide intelligent help to the user.

10

15

20

### **§ 2. SUMMARY OF THE INVENTION**

25

The present invention may represent, visually, objects (such as a document, a spread sheet, a business contact, a drawing, a picture or image, a web page, a resource location or directory, etc., or a representation

thereof, for example) or content with a low resolution scaled image (such as, a 64 pixel by 64 pixel bit map having 24 bit color for example) which can be added, moved, or deleted from a display rendered on a video monitor. The present invention may use pop-up title bars (or other descriptive textual information) for permitting a user to discern more about the object represented by a low resolution image. The present invention may use higher resolution image (such as a 512 pixel by 512 pixel bit map having 24 bit color for example) representations of objects, or "live" objects loaded within an associated application, for editing or otherwise working on a selected object. Other visual representations of objects may be used. Any and all visual representations of objects may be referred to as "thumbnails" in the following.

The present invention may render a two-dimensional surface (or layer) on which the object thumbnails may be manipulated. The two-dimensional surface upon which an object may be manipulated may be the surface of a simulated three-dimensional tank (also referred to as a "data tank").

The present invention may map values of properties or parameters of the object information to a depth. Using this depth information, the present invention renders the thumbnail of the information object

in the simulated three-dimensional tank at a simulated depth. As the values of properties or parameters of the objects change, the simulated depth at which they are rendered may change.

5

The present invention may use inputs from a familiar input device such as a mouse or pointer to manipulate the object thumbnails. The present invention may also prevent any one object thumbnail (or landscape feature) from totally occluding another object thumbnail so that neither is (totally) hidden from the user.

10

15

20

25

Finally, the present invention may cluster, by rendering a visual boundary for example, object thumbnails which are arranged, by the user, relatively close to one another. Further, the present invention may employ a matching (e.g., correlation or similarity) algorithm to determine whether certain objects are related. Alternatively, objects may be related by an explicit selection or designation by a user. If so, the present invention may employ some type of visual indication, such as a colored halo around thumbnails of related objects for example, of related objects. This visual indication may be rendered continuously or, alternatively, upon an event, such as when an object thumbnail is made "active".

### § 3. BRIEF DESCRIPTION OF THE DRAWINGS

The file of this patent contains at least one drawing executed in color. Copies of this patent with color drawing(s) will be provided by the Patent and Trademark Office upon request and payment of the necessary fee.

Figure 1 is a personal computer on which the user interface of the present invention may be effected.

Figure 2A is a high level diagram of processes and stored data which may be used to effect the user interface of the present invention. Figure 2B is a diagram of processes which may be related to an input management process of Figure 2A. Figure 2C is a diagram of processes which may be related to a state update management process of Figure 2A. Figure 2D is a diagram of processes which may be related to a special services management process of Figure 2A. Figure 2E is a diagram of processes which may be related to an output management process of Figure 2A.

Figure 3 is an exemplary data structure of data which may be stored and used when effecting the user interface of the present invention.

Figure 4 is an exemplary data structure of a record of low resolution image data of an object which may be used when effecting the user interface of the present invention.

5

Figure 5 is an exemplary data structure of a record of high resolution image data of an object which may be used when effecting the user interface of the present invention.

10

Figure 6 is an exemplary data structure of a record of title (or other) data which may be used when effecting the user interface of the present invention.

15

Figure 7 is an exemplary data structure of a record of implicit query information which may be used when effecting the user interface of the present invention.

20

Figure 8 is an exemplary data structure for storing animations which may be used by the present invention.

25

Figure 9 is an exemplary display of an output of the user interface of the present invention.



Figures 10A through 10C depict a sequence of exemplary displays which show the affects of selecting objects in the user interface of the present invention.

5                Figures 11A and 11B depict a sequence of exemplary displays which show a "diving" feature which may be provided by the user interface of the present invention.

10              Figure 12 is an exemplary display of an output of the user interface of the present invention with a non-plan viewing point.

15              Figure 13 is a high level flow diagram of an animation loop process which may be used by the user interface of the present invention.

20              Figure 14 is a flow diagram of an exemplary method for performing an input management process.

                Figure 15 is a flow diagram of an exemplary method for performing a state update management process.

25              Figure 16 is a flow diagram of an exemplary method for performing an object status and location update process.

Figure 17 is a diagram of states and state transitions which the user interface of the present invention may use.

5                Figures 18A and 18B are flow diagrams of alternative object occlusion avoidance methods.

10              Figure 19 is a flow diagram of an exemplary method for performing an object depth and fade or shade determination process.

15              Figure 20 is a flow diagram of an exemplary method for performing a view point determination process.

20              Figure 21 is a flow diagram of an exemplary method for performing an output management process.

25              Figure 22 is a machine on which the user interface of the present invention may be effected.

#### **§ 4. DETAILED DESCRIPTION**

The present invention concerns novel methods, apparatus and data structures for providing a user interface. The following description is presented to enable one skilled in the art to make and use the invention, and is provided in the context of particular applications and their requirements. Various

modifications to the disclosed embodiments will be apparent to those skilled in the art, and the general principles set forth below may be applied to other embodiments and applications. Thus, the present invention is not intended to be limited to the embodiments shown.

Features of, and functions which may be performed by, the present invention, will first be described in § 4.1 below. Then, structures, processes, data structures, methods and displays of exemplary embodiments of the present invention will be described in § 4.2 below.

#### **§ 4.1      FUNCTIONS WHICH MAY BE PERFORMED BY THE PRESENT INVENTION**

Recall from § 1.2.3 above, that there exists a need for a user interface, and in particular a graphical user interface, to information or content (also referred to as an "object"). A user should be able to view and organize all objects and to edit or otherwise work on a selected object. To achieve these goals, the present invention may represent, visually, objects (such as a document, a spread sheet, a business contact, a drawing, a picture or image, a web page, a resource location or directory, etc., or a representation thereof, for example) or content with a low resolution scaled image (such as, a 64 pixel by 64 pixel bit map having 24 bit

color for example) which can be added, moved, or deleted from a display rendered on a video monitor. The present invention may use pop-up title bars (or other descriptive textual information) for permitting a user to discern  
5 more about the object represented by a low resolution image. The present invention may use higher resolution image (such as a 512 pixel by 512 pixel bit map having 24 bit color for example) representations of objects, or "live" objects loaded within an associated application,  
10 for editing or otherwise working on a selected object. Other visual representations of objects may be used. Any and all visual representations of objects may be referred to as "thumbnails" in the following.

15 Recall also from § 1.2.3 above, that a user interface should exploit spatial memory. To achieve this goal, the present invention may render a two-dimensional surface (or layer) on which the object thumbnails may be manipulated. The two-dimensional surface upon which an  
20 object may be manipulated may be the surface of a simulated three-dimensional tank (also referred to as a "data tank").

25 Recall also from § 1.2.3 above that a user interface should convey, at minimal cognitive load to the user, properties or parameters of the information to the user. To achieve this goal, the present invention may map values of properties or parameters of the object

information to a depth. Using this depth information,  
the present invention renders the thumbnail of the  
information object in the simulated three-dimensional  
tank at a simulated depth. As the values of properties  
5 or parameters of the objects change, the simulated depth  
at which they are rendered may change.

Recall further from § 1.2.3 above that a user  
interface should also be intuitive to minimize the time  
10 needed for a user to become familiar with it. For  
example, the user interface should not be cumbersome, nor  
should it require too many different types of inputs to  
manipulate the object thumbnails. To achieve this goal,  
the present invention may use inputs from a familiar  
15 input device such as a mouse or pointer to manipulate the  
object thumbnails. The present invention may also  
prevent any one object thumbnail (or landscape feature)  
from totally occluding another object thumbnail so that  
neither is (totally) hidden from the user.

20 Recall finally from § 1.2.3 above, that the  
user interface should provide intelligent help to the  
user. The present invention may cluster, by rendering a  
visual boundary for example, object thumbnails which are  
25 arranged, by the user, relatively close to one another.  
Further, the present invention may employ a matching  
(e.g., correlation or similarity) algorithm to determine  
whether certain objects are related. Alternatively,

objects may be related by an explicit selection or designation by a user. If so, the present invention may employ some type of visual indication, such as a colored halo around thumbnails of related objects for example, of related objects. This visual indication may be rendered continuously or, alternatively, upon an event, such as when an object thumbnail is made "active".

Having described functions which may be performed by the present invention, structures, processes, methodologies, data structures and displays which may be used by the present invention are now described in § 4.2.

#### **§ 4.2 STRUCTURES, PROCESSES, METHODOLOGIES, DATA STRUCTURES, AND DISPLAYS WHICH MAY BE USED BY THE PRESENT INVENTION**

In the following, exemplary systems on which the present invention may operate are described in § 4.2.1 with reference to Figures 1A and 1B, exemplary displays which may be generated by the present invention are described in § 4.2.2 with reference to Figures 9 through 12, and exemplary processes, methods and data structures which may be used to effect certain aspects of the present invention are described in § 4.2.3 with reference to Figures 2A through 8 and 13 through 21.

**§ 4.2.1 EXEMPLARY SYSTEMS (FIGS. 1 and 22)**

Figure 1 and the following discussion provide a brief, general description of an exemplary apparatus in which at least some aspects of the present invention may be implemented. The present invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. However, the methods of the present invention may be effected by other apparatus. Program modules may include routines, programs, objects, components, data structures, etc. that perform a task(s) or implement particular abstract data types. Moreover, those skilled in the art will appreciate that at least some aspects of the present invention may be practiced with other configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network computers, minicomputers, set top boxes, mainframe computers, and the like. At least some aspects of the present invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices linked through a communications network. In a distributed computing environment, program modules may be located in local and/or remote memory storage devices.

With reference to Figure 1, an exemplary apparatus 100 for implementing at least some aspects of the present invention includes a general purpose computing device in the form of a conventional personal computer 120. The personal computer 120 may include a processing unit 121, a system memory 122, and a system bus 123 that couples various system components, including the system memory 122, to the processing unit 121. The system bus 123 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory may include read only memory (ROM) 124 and/or random access memory (RAM) 125. A basic input/output system 126 (BIOS), containing basic routines that help to transfer information between elements within the personal computer 120, such as during start-up, may be stored in ROM 124. The personal computer 120 may also include a hard disk drive 127 for reading from and writing to a hard disk, (not shown), a magnetic disk drive 128 for reading from or writing to a (e.g., removable) magnetic disk 129, and an optical disk drive 130 for reading from or writing to a removable (magneto) optical disk 131 such as a compact disk or other (magneto) optical media. The hard disk drive 127, magnetic disk drive 128, and (magneto) optical disk drive 130 may be coupled with the system bus 123 by a hard disk drive interface 132, a magnetic disk drive interface 133, and a (magneto)



optical drive interface 134, respectively. The drives and their associated storage media provide nonvolatile (or persistent) storage of machine readable instructions, data structures, program modules and other data for the personal computer 120. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 129 and a removable optical disk 131, those skilled in the art will appreciate that other types of storage media, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROM), and the like, may be used instead of, or in addition to, the storage devices introduced above.

A number of program modules may be stored on the hard disk 127, magnetic disk 129, (magneto) optical disk 131, ROM 124 or RAM 125, such as an operating system 135 (for example, Windows NT® 4.0, sold by Microsoft Corporation of Redmond, Washington), one or more application programs 136, other program modules 137 (such as "Alice", which is a research system developed by the User Interface Group at Carnegie Mellon University available at [www.Alice.org](http://www.Alice.org), OpenGL from Silicon Graphics Inc. of Mountain View California, or Direct 3D from Microsoft Corp. of Bellevue Washington), and/or program data 138 for example.

A user may enter commands and information into the personal computer 120 through input devices, such as a keyboard 140, a camera 141 and pointing device 142 for example. Other input devices (not shown) such as a microphone, joystick, game pad, satellite dish, scanner, or the like may also be included. These and other input devices are often connected to the processing unit 121 through a serial port interface 146 coupled to the system bus. However, input devices may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). For example, since the bandwidth of the camera 141 may be too great for the serial port, the video camera 141 may be coupled with the system bus 123 via a video capture card (not shown). The video monitor 147 or other type of display device may also be connected to the system bus 123 via an interface, such as a video adapter 148 for example. The video adapter 148 may include a graphics accelerator. One or more speaker 162 may be connected to the system bus 123 via a sound card 161 (e.g., a wave table synthesizer such as product number AWE64 Gold Card from Creative Labs of Milpitas, California). In addition to the monitor 147 and speaker(s) 162, the personal computer 120 may include other peripheral output devices (not shown), such as a printer for example. As an alternative or an addition to the video monitor 147, a stereo video output device, such as a head mounted display or LCD shutter glasses for example, could be used.

The personal computer 120 may operate in a networked environment which defines logical connections to one or more remote computers, such as a remote  
5 computer 149. The remote computer 149 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and may include many or all of the elements described above relative to the personal computer 120, although only a memory storage  
10 device has been illustrated in Figure 1A. The logical connections depicted in Figure 1A include a local area network (LAN) 151 and a wide area network (WAN) 152, an intranet and the Internet.

15 When used in a LAN, the personal computer 120 may be connected to the LAN 151 through a network interface adapter (or "NIC") 153. When used in a WAN, such as the Internet, the personal computer 120 may include a modem 154 or other means for establishing  
20 communications over the wide area network 152. The modem 154, which may be internal or external, may be connected to the system bus 123 via the serial port interface 146. In a networked environment, at least some of the program modules depicted relative to the personal  
25 computer 120 may be stored in the remote memory storage device. The network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Figure 22 is a more general machine 2200 which may effect one or more of the processes discussed above. The machine 2200 basically includes a processor(s) 2202, an input/output interface unit(s) 2204, a storage device(s) 2206, and a system bus or network 2208 for facilitating data and control communications among the coupled elements. The processor(s) 2202 may execute machine-executable instructions to effect one or more aspects of the present invention. At least a portion of the machine executable instructions and data structures may be stored (temporarily or more permanently) on the storage devices 2206 and/or may be received from an external source via an input interface unit 2204.

#### **§ 4.2.2 EXEMPLARY DISPLAYS (FIGS. 9-11B)**

As discussed in § 4.1 above, present invention may represent, visually, objects using object thumbnails. These thumbnails may be manipulated on a two-dimensional surface and may have a simulated depth based on values of properties or parameters of the object. Figure 9 is a display 900 which illustrates the data tank metaphor user interface of the present invention. Basically, the display 900 includes an object manipulation and display field 910 and a view point control panel 920. Like a glass-walled tank, the object manipulation and display field 910 includes a bottom 912, a front wall 914, a rear

wall 916, and side walls 918a and 918b. An equally valid metaphor is that of looking into a room (not a glass-walled tank) which includes a back wall 912, a floor 914, a ceiling 916 and side walls 918a and 918b.

5 The front wall is transparent, giving the entire mechanism the visual appearance of a stage or a proscenium.

10 The control panel 920 includes buttons, which may be actuated using a pointing device, such as a mouse for example. The height control sub-panel 930 permits the user's viewing point to move up-and-down in directions perpendicular to the surface of the tank. Button 931 moves the viewing point up (the tank will appear to move lower on the screen) and button 932 moves the viewing point lower (making the tank move appear to move higher).  
15 Button 933 will return the viewing point to a "home" position, such as the top surface of the tank for example. A tilt control sub-panel 940 permits a user's viewing point be rotated about a tilt axis 943, in direction 943a using button 942, or in direction 943b using button 941. Buttons 952 and 953 permit a user's viewing point to be translated left and right, respectively. Panel 951 permits a user's viewing point  
20 to be moved forward and rearward, (for example, assuming that there has been no net rotation about the tilt axis 943, toward the bottom of the tank and toward the top surface of the tank, respectively) or turned from one  
25

side to the other in a motion similar to camera pan on a standard camera tripod.

5           The height button 932 or (assuming that there  
has been no net rotation about the tilt axis 943) the  
forward button 951a of panel 951 allows the user's  
viewing point to descend into the (also referred to as  
"diving") while the height button 931 or (assuming that  
there has been no net rotation about the tilt axis 943)  
10   the backward button 951b of panel 951 permits the user's  
viewing point to ascend to the surface again. For  
example, referring to Figure 12, the viewing point is  
located in front of the front wall 914 of the tank and to  
the right of the right wall 918b of the tank. Finally, a  
15   reset camera button 954 may be used to return a user's  
viewing point to a home position, such as centered  
between the side walls 918a, 918b and located over the  
top surface of the tank, as shown in Figure 9, with no  
rotation about the tilt axis 943. In an alternative  
20   embodiment, "diving" may be effected in response to a  
mouse click on an visible part of the tank and a drag on  
the y-axis (depth). Similarly, different parts of the  
tank (e.g., floor, walls, etc.) can be grabbed and  
dragged along different axes to change the viewing point  
25   location.

Figures 10A through 10C depict a sequence of  
exemplary displays which show the affects of "floating"

objects in the user interface of the present invention. Referring first to the display 1000 of Figure 10A, notice that three (3) thumbnails 1010, 1020, and 1030 are provided in the data tank. Notice that each of the  
5 thumbnails may project a shadow 1040 onto the floor 912 of the tank. Notice also that as the depth of the thumbnails increases, the thumbnails may be increasingly shaded or faded out. Specifically, in the exemplary display shown, the thumbnails are being tinted toward the  
10 blue end of the spectrum using the fog-rendering capabilities of the 3D graphics system. Finally, notice that the perceived sizes of the thumbnails are different since they are rendered from the perspective of the viewing point which, in this case, is centered above the  
15 tank.

The scene is rendered in a 3D perspective projection; objects further away from the view point appear (are rendered) smaller on the screen. This is in  
20 contrast to parallel projection 3D graphics systems in which objects do not get smaller with distance. The present invention could use parallel projection. However, if perspective projection is not used, the user loses an important depth cue.

25 In display 1000' of Figure 10B, the thumbnail 1020 has been "floated" (for example, by left clicking a mouse button when a mouse controlled cursor is

over the thumbnail 1020). Thus, this thumbnail 1020 is brought to the surface of the tank where it can be moved, using traditional "drag and drop" inputs for example. Notice that since this thumbnail 1020 is now at the surface of the tank, it is not shaded or faded.

In display 1000'' of Figure 10C, the thumbnail 1030 has been "floated" (for example, by left clicking a mouse button when a mouse controlled cursor is over the thumbnail 1030). Thus, this thumbnail 1030 is brought to the surface of the tank where it can be moved, using traditional "drag and drop" inputs for example. Notice that since this thumbnail 1030 is now at the surface of the tank, it is not shaded or faded. Notice also that the thumbnail 1020, which was previously on the surface of the tank in display 1000' of Figure 10b, has started to "sink" in the tank. In this case, the depth may be a linear function of the number of mouse clicks since the thumbnail was last "floated".

Recall that the button 932 or (assuming that there has been no net rotation about tilt axis 943) the forward button 951a of the panel 951 permits a user's viewing point to descend into the tank (also referred to as "diving"), and that the button 931 or (assuming that there has been no net rotation about tilt axis 943) the backward button 951b of the panel 951 allows the viewing point to ascend within the tank. Button 933 allows the



viewpoint to instantaneously return to a home viewing point, such as viewing the top surface of the tank. Figures 11A and 11B depict a sequence of exemplary displays which show a viewing point descent (or "diving") feature which may be provided by the user interface of the present invention.

Referring first to Figure 11A, a display 1100 includes a tank having a number of thumbnails 1102, 1104, 1106, 1108, 1110, 1112, 1114, 1116, 1118, 1120, 1122, and 1124. As was the case with the displays of Figures 10A through 10C above, notice that (i) each of the thumbnails may project a shadow onto the floor 912 of the tank, (ii) as the depth of the thumbnails increases, the thumbnails may be increasingly shaded or faded out, and (iii) the perceived sizes of the thumbnails is different since they are rendered from the perspective of the viewing point.

Now, assume that the user activated the button 932 or the forward button 951a of the panel 951. Referring to the display 1100' of Figure 11B, notice that viewing point has descended within the tank such that objects 1112, 1114, 1116, 1118, 1120, 1122, and 1124 are no longer visible since they are "above" the viewing point. This is similar to the known concept of "culling" of objects which occurs when objects are "behind" the viewing point. However, even these thumbnails may still cast shadows to the floor 912 of the tank.

Alternatively, thumbnails "above" the viewing point need not cast shadows in the manner described. Comparing Figures 11A and 11B, due to the change in the viewing point, the perceived sizes of the thumbnails 1102, 1104, 1106, 1108, and 1110, as well as the floor 912 of the tank, are larger in display 1100' than in display 1100. However, the shading or fading of the thumbnails is the same. Alternatively, as the user's viewing point "dives" into the tank, the object thumbnails may become clearer.

Finally, recall from § 4.1 above that the present invention may use higher resolution image representations of objects, or "live" objects within an associated application, for editing or otherwise working on a selected object. The selected object thumbnail may be displayed in a preferred viewing position. The selected object thumbnail may be a high resolution bit map (e.g., 512 pixels by 512 pixels with 24 bit color). To reiterate, rather than merely providing a high resolution object thumbnail, the actual object, in its associated application, may be presented. In this example, the Internet Explorer™ Internet browser (part of Microsoft Windows NT® 4.0 operating system sold by Microsoft Corporation of Redmond, Washington) may be rendering a web page, with the user interface of the present invention in the background. The application can be maximized, as is known to those skilled in the art, to substantially fill the screen of the video monitor.

Further, the images may be rendered as an off-screen  
bitmap, which can be placed in the preferred viewing  
position when an object is selected. If the objects are  
HTML web pages, this may be done, for example, using  
5 techniques discussed in the article, M. Heydt,  
"Incorporating the Web Browser Control Into Your  
Program", Microsoft Interactive Developer, Vol. 3, No. 7,  
pp. 42-51 (July 1998).

10 In consideration of interface consistency to  
the user, when an object thumbnail not on the surface of  
the tank is "floated", an animation, taking on the order  
of one second, may be used to move the object thumbnail  
from its position and depth, to a corresponding position  
15 on the surface of the tank. Similarly, when an object  
thumbnail, below, or on, the surface of the tank is  
"selected", an animation, taking on the order of one  
second, may be used to move the object thumbnail from its  
position below, or on, the surface of the tank to a  
20 preferred viewing location (such as above the surface of  
the tank) and use the larger high resolution thumbnail or  
the application with the "live" object. In each case,  
the user will perceive the object thumbnail as moving  
towards them.

25 Having described functions which may be  
performed by the present invention, as well as exemplary  
displays which may be rendered by the present invention,

exemplary processes, methods, and data structures which may be used to effect, at least certain aspects of the present invention, as now described in § 4.2.3.

**§ 4.2.3 EXEMPLARY PROCESSES, METHODS AND DATA STRUCTURES**

In the following, a high level overview of processes and an animation loop method will be presented in § 4.2.3.1 with reference to Figures 2A, 3 through 8, and 13. Thereafter, processes and methods for managing inputs, updating state information, managing special services, and managing outputs, will be presented in § 4.2.3.2 through § 4.2.3.5.

**§ 4.2.3.1 OVERVIEW OF PROCESSES AND ANIMATION LOOP METHOD**

In the following, high level processes which may be used to effect at least some aspects of the user interface of the present invention, are described, with reference to Figure 2A, in § 4.2.3.1.1. Next, exemplary data structures for storing data which may be used by one or more processes of the present invention are described, with reference to Figures 3 through 8, in § 4.2.3.1.2. Finally, an exemplary high level method for effecting at least some aspects of the user interface of the present invention is described, with reference to Figure 13, in § 4.2.3.1.3.

**§ 4.2.3.1.1      HIGH LEVEL PROCESSES  
(FIGURE 2A)**

Figure 2A is a high level diagram of processes and stored data which may be used to effect, at least certain aspects of, the user interface of the present invention. Basically, the processing by the present invention may be thought of as a sequence of cycles. In each cycle, user inputs are accepted, states are updated, if necessary, based on such user inputs, and a display is rendered (and audio is output) based on the updated states. More specifically, referring to Figure 2A, user inputs from input devices 210 are managed by an input management process (or more generally, an "input manager") 220. A state update management process 230 uses information from the input management process 220, as well as information of a previous state stored at temporary storage means 240, to update the state. This updated state is then stored back to the temporary storage means 240. To the extent that state changes or transitions need to be accounted for, the temporary storage means 240 may also store one (or more) preceding state. Alternatively, state change or state transition flags may be stored in the temporary storage means 240. A special services management process 250 may be used to perform special functions, such as proximity clustering and implicit query. Finally, based on the state information stored in the temporary storage means 240, and information which may be generated by the special

services management process, an output management process (or more generally, an "output manager") 260 generates outputs (such as video display and audio signals) for provision to the output devices 270. Persistent storage 280 can store information which may be used and/or generated by the state update management process 230, the special services management process 250, and/or the output management process 260. The timer/clock facility 290 may be used to provide relative timing functions and absolute times to any of the processes.

**§ 4.2.3.1.2      EXEMPLARY DATA  
STRUCTURES  
(FIGURES 3-8)**

Below, the state information, which may be stored in the temporary storage device 240, is described. Figure 3 is an exemplary data structure 300 of state (and other) information which may be used by the user interface of the present invention. Object information 302 may include a record 304 corresponding to each object. As shown in Figure 3, an object record 304 may include an object identifier field 306 which includes a value used to distinguish the object from all other objects. For example, if the object is an HTML page on the Internet, the object identifier may be a URL (or "Uniform Resource Locator"). Accordingly, each object should have a unique object identifier value in field 306. The object record 304 may also include a

location field 308 which contains information about the location of the object (or object thumbnail). The location information in field 308 may include a two-dimensional plane location and a depth, or coordinates which may be mapped to a screen location. In one embodiment, a grid (not displayed to the user) may be used to overlay the monitor screen or the layers of depth of the data tank. In this case, for each layer of depth, each object will be anchored to a particular grid coordinate which will be stored in that object's location field 308. If an object thumbnail is "pushed", as will be discussed in detail below to avoid full object occlusion, in one embodiment, its pre-push location also may be stored in field 310 for reasons which will become apparent below. Each object record 304 may also include a field 312 containing a flag indicating whether or not the object is "active" (as described below), a field 313 containing a flag indicating whether or not the object is "floated", a field 314 containing a flag indicating whether or not the object is "selected", a field 316 containing a flag indicating whether or not the object is "moving" (as described below), a field 318 containing a flag indicating whether or not the object is being "pushed" (as described below), and a field 320 containing a fade or shade parameter. Regarding field 320, since a fade or shade parameter may be determined on the basis of the object's depth, this field may include redundant information which might not be strictly necessary. If

any of the objects are selected, the "any page selected?" flag is set in field 322.

In addition to the object information 302, the state information 300 may also includes a cursor location field 324 for storing a two-dimensional location of a cursor. The number of objects may be stored in field 326. The viewing point location (and orientation) may be stored in field 360.

Other parameters may be used when rendering the display. These parameters are either fixed or may have default values which may be changed by the user. For example, a tank content may be fixed, or may be selected or changed by the user and stored in field 328. For example, if the tank content is "water", the thumbnails may be rendered darker and shaded more blue as their depth within the tank increases. If the tank content is "fog", the thumbnails may be rendered with less resolution and/or with darker shading as their depth increases. Although the color and/or shading are shown as blue in the exemplary displays, alternatively, other colors, or even a grayscale, may be used to provide a visual depth cue. The type of layout (such as gridded or continuous with push, as will be described later) may be fixed, or may be changed or selected by the user and stored in field 338. If the layout is a gridded layout, a grid map, which defines the resolution of the grid, may



be fixed, or may be changed or selected by the user and stored in field 336. If the invention is to include an implicit query function, (as will be described later), a "match threshold" value may be fixed, or may be selected or changed by a user and stored in field 340. Finally, depth parameter(s), as well as a parameter to depth mapping function, may be stored in fields 352 and 350, respectively. For example, the depth may be defined as:

DEPTH = number of clicks since last selection.

Thus, depth may be a linear function of a single parameter. Alternatively, depth may be defined as:

DEPTH = log (number of clicks since last selection).

Thus, the depth may be a logarithmic function of a single parameter. Basically, the depth of an object may be a linear, polynomial, or exponential function of one or more of its parameters or properties. Thus, for example, an object's depth may be defined as:

DEPTH = (size in bytes / 10) x (number of clicks since last selection)<sup>0.75</sup>.

**§ 4.2.3.1.3      EXEMPLARY METHOD  
(FIGURE 13)**

Figure 13 is a high level flow diagram of an  
5 exemplary animation loop process 1300. First, as shown  
in act 220'/1310, an input is accepted. This act will be  
discussed later with reference to Figure 14. Next, as  
shown in act 230'/1320, states are updated based on  
previous states and accepted inputs. This act will be  
10 discussed later with reference to Figures 15 through 20.  
Next, as shown in act 1330, the updated states are  
stored. (Recall, for example, the temporary  
storage 240.) As shown in optional act 250'/1340,  
special services, such as launching an implicit query or  
15 clustering proximally located thumbnails, may be carried  
out. Finally, in act 260'/1350, a display is rendered  
and audio is output based on the updated states. This  
act will be discussed later with reference to Figure 21.  
At conditional branch point 1360, it is determined  
20 whether or not a exit command was entered by the user.  
If so, the method 1300 is left via RETURN node 1370. If,  
on the other hand, no exit command was entered by the  
user, processing branches back to act 220'/1310.

**§ 4.2.3.2      MANAGING INPUTS**

25 In the following, processes related to managing  
inputs are described, with reference to Figure 2B, in  
§ 4.2.3.2.1. Thereafter, an exemplary method for

performing this process is described, with reference to Figure 14, in § 4.2.3.2.2.

**§ 4.2.3.2.1      PROCESSES FOR MANAGING  
INPUTS (FIGURE 2B)**

Processes related to the input management process 220 are now described with reference to Figure 2B. As shown in Figure 2B, a user may interact with the user interface of the present invention using a pointing device 210a, such as a mouse for example. The pointer input management process (or more generally, a "2D input facility") 212 provides user inputs, from the pointing device 210a, to the input management process 220, either directly or through an event queue 216. A user may also interact with the user interface of the present invention using gestures captured by a camera 210b. The camera input management process (or more generally, a "camera input facility") 214 provides user inputs, from the camera 210b, to the input management process 220, either directly or through an event queue 216. The input management process 210 passes this information to a state update management process 230.

**§ 4.2.3.2.2      EXEMPLARY METHOD FOR  
MANAGING INPUTS  
(FIGURE 14)**

5            Recall from act 220'/1310 that inputs are  
accepted. Figure 14 is a high level flow diagram of an  
exemplary method 220' for effecting the input management  
process. As shown in act 1410, a next event is fetched  
from a queue. The method 220' is then left via RETURN  
10 node 1420.

**§ 4.2.3.3      MANAGING STATE INFORMATION**

15           In the following, processes related to managing  
state information are described, with reference to  
Figure 2C, in § 4.2.3.3.1. Thereafter, exemplary methods  
for performing these processes are described, with  
reference to Figures 15 through 20, in § 4.2.3.3.2.

20           **§ 4.2.3.3.1      PROCESSES FOR MANAGING  
STATE INFORMATION  
(FIGURE 2C)**

25           Processes related to the state update  
management process 230 are now described with reference  
to Figure 2C. Referring back to Figure 2A, the state  
update management process 230 may accept information from  
the input management process 220. The cursor location  
determination process (or more generally, a "cursor  
30 location determination facility") 237 can use

information, originally from the mouse/pointer 210a, as well as present cursor position information (Recall field 324 of Figure 3.), to determine a present location of a cursor. This may be done in a known manner for example.

The view point determination process (or more generally, a "view point determination facility") 235 may use cursor location information, information, originally from the camera 210b, as well as present view point information (Recall field 360 of Figure 3.) to determine a present view point and orientation. Figure 20 is a flow diagram of an exemplary method 235' for performing a view point determination process. First, as shown in act 2010, control panel (Recall, e.g., sub-panels 930 and 940, and buttons 951-954 of Figure 9.) and camera inputs are accepted. Next, as shown in conditional branch point 2020, if a pan command exists (Recall buttons 952 and 953 of Figure 9.), the view point is updated accordingly as shown in act 2030 and processing continues to conditional branch point 2040. Returning back to conditional branch point 2020, if a pan command does not exist, then processing continues directly to conditional branch point 2040. As shown in conditional branch point 2040, if a dive (or ascend) command exists (Recall, e.g., sub-panel 951 of Figure 9.), the view point is updated accordingly as shown in act 2050 and processing continues to conditional branch point 2070. Returning

back to conditional branch point 2040, if a dive (or ascend) command does not exist, then processing continues to act 2060 where the viewing point is "floated" by a predetermined amount (not necessarily all the way to the surface). Alternatively, act 2060 may be skipped. At conditional branch point 2070, if a tilt command exists (Recall, e.g., sub-panel 940 and buttons 941 and 942.), the viewing point is updated accordingly as shown in act 2080 and processing continues to RETURN node 2090. Returning to conditional branch point 2070, if a tilt command is not determined, then processing continues directly to the RETURN node 2090.

Although the pan, dive and tilt commands were described with reference to sub-panels or buttons, head gestures of a user, captured by the camera 210b and interpreted can be used instead of, or in addition to, such input means. For example, a user leaning their head towards or away from the camera may be interpreted as diving and ascending. A user turning their head left, right, up, or down may be interpreted as tilting. Finally, a user moving their head left, right, up, or down may be interpreted as panning. For example, the camera inputs may be processed to determine whether the user's head is located to one side, or the other, or above or below the center of the video monitor. The position of the user's head may be determined using known techniques such as detecting a "blob" of color as a face

or a "draping" technique in which an image from the camera before the user was seated before the monitor is subtracted from a current image.

5                   Referring back to Figure 2C, using the location of the cursor, as well as the locations and depths of the objects stored in fields 308 of records 304 of the object information 302, the object state and location determination process (or more generally, an "object state update facility") 231 may determine, for each  
10                   object: (i) whether the object is "active", "floated", "selected", "moving", or "pushed"; (ii) the location of the object; and (iii) the depth of the object. If an object is "moving", its location is updated. Further, if  
15                   any object is "moving", an (total) occlusion avoidance process 233 process may be performed. Each of the object states is now described with reference to Figure 17.

                  Figure 17 is a diagram of states and state  
20                   transitions which the user interface of the present invention may use. Starting from an initial or default state 1700, if an object is activated, the object becomes "active" at state 1710. An object may become activated, for example, when a cursor is on, or passes over, or  
25                   hovers over (note timer/clock facility 290.) an object thumbnail. The object may be "deactivated" when the cursor is taken off the object for example, in which case, the default state 1700 is reentered.

Alternatively, if an active object is below the tank surface and a float command, for example a mouse click, is issued, the object becomes "floated" at state 1715. If the object is "sunk", for example by a right mouse click, the state 1710 in which the object is "active" is reentered. Alternatively, if a "floated" object is selected, for example, by a single mouse click, the object "selected" state 1720 is entered. In one embodiment, any object (that is, not necessarily only a "floating" object) may enter the object "selected" state 1720 when a selection command, such as a double mouse click for example, is entered. In the object "selected" state, an enlarged high resolution thumbnail of the object may be rendered at a preferred viewing location, such as at the bottom center of the display screen for example. Alternatively, in the object "selected" state, a "live" object may be loaded into its associated application, such as a WORD document being rendered in a word processing program or an HTML page being rendered on an Internet browser program, for example. From the object "selected" state 1720, the object "floated" state 1715 can be reentered in response to a de-selection command, such as a mouse click on a "minimize" button for example.

Referring back to the "active" object state 1710, if the object is subject to a translation command, for example via a traditional left mouse button



dragging operation, the "moving" state 1730 is entered. From the "moving" state 1730, if the object is released, for example by releasing the left mouse button during a drag operation, the "active" state 1710 is reentered.

5 When the object is in the "moving" state 1730, its location will be updated. Object occlusion may be avoided as follows. In one embodiment (not shown), only "floated" objects may be moved. In yet another embodiment, any object, including floating objects, may  
10 be moved.

In the following descriptions of object occlusion avoidance techniques, only objects at the same depth may be considered or, alternatively, all objects  
15 may be projected to a predetermined depth, such as the surface of the data tank for example. Note that in the latter alternative, the depths of the objects are basically ignored. In yet another alternative, one or more depths may be grouped and considered together for  
20 the purposes of object occlusion avoidance. For example, if an object is being moved, objects in the five depth levels above and in the five depth levels below that object may be considered during object occlusion avoidance.

25 With the foregoing alternatives in mind, assuming that the layout type (Recall field 338) is "continuous with push", if the location or position of

the moving object is within a predetermined distance of another object (at its depth layer, within a predetermined number of depth layers, or at any depth), the other object enters the "pushed" state 1740 and its location is updated so that (total) occlusion between the "moving" object and the "pushed" object is avoided. Note that a "pushed" object may cause other objects to be pushed. In one embodiment, if the moving object is no longer within the predetermined distance of the pre-push location of the other object, the other object is no longer "pushed" and takes on its pre-push location as shown by the dashed transition lines and state 1745. An animation may be used to move a pushed object from a current location to its pre-push location. The animation may take on the order of one (1) second. In one embodiment, the animation takes less than 1/2 second. In another embodiment, the distance between the current location of the "pushed" object and the moving object is used to determine whether a "pushed" object continues to be "pushed". If, while a "moving" object is pushing a "pushed" object, the "moving" object is no longer pushed or pulled, for example, by releasing the left mouse button during a drag operation, the default state 1700 is reentered and objects previously "moved" and "pushed" are no longer in those respective states, and they are rendered at their last determined locations. If the object "moving" takes on a location beyond the

predetermined distance of the other object being  
"pushed", that other object will no longer be "pushed".

The foregoing assumed that the layout type was  
5 "continuous with push". Assuming, on the other hand,  
that the layout type is "gridded", an object may be  
continuously moved about the landscape. However, when  
the object is released, its position assumes the nearest  
open (i.e., not taken by another object) grid position.  
10 To reiterate, the grid may overlay the screen or any  
layer of the tank. The resolution of the grid is stored  
in the grid map field 336.

Recall that a tank content may be fixed, or may  
15 be selected or changed by the user and stored in  
field 328. Recall, for example, that if the tank content  
is "water", the thumbnails may be rendered darker and  
shaded more blue as their depth increases. If the tank  
content is "fog", the thumbnails may be rendered with  
20 less resolution and/or shaded darker as their depth  
increases. Referring back to Figure 2C, a fade or shade  
determination process 232 may be used to determine and  
change a display characteristic (such as shade,  
resolution, darkness, etc.) of an object thumbnail based  
25 on its depth.

**§ 4.2.3.3.2      EXEMPLARY METHODS FOR  
MANAGING STATE  
INFORMATION**

5            Recall from act 230'/1320 that states are  
updated based on previous states and accepted inputs.  
Figure 16 is a flow diagram of an exemplary method 230'  
for effecting a state update management process. First,  
as shown in act 237'/1510, a cursor location is  
10            determined. This determination may be based on a  
previous cursor location and pointer inputs, and may be  
performed in a known way. Further, as shown in  
act 235'/1520, a view point location and orientation is  
determined. This determination may be based on previous  
15            view point information and pointer inputs relative to the  
control panel 920. Next, as shown in act 231'/1530, the  
status and location of an object is updated, if  
necessary. An exemplary method for performing this act  
will be described below with reference to Figure 16.  
20            Finally, as shown in act 232'/1540, for each object, a  
depth and fade or shade is determined. An exemplary  
method for performing this act will be described below  
with reference to Figure 19.

25            As just described, a state update management  
process 230 may include a act of updating, if necessary,  
an object's status and location. Figure 16 is a flow  
diagram of an exemplary method 231'/1530 for effecting an  
object status and location update process. First, at

conditional branch point 1605, it is determined whether or not the cursor is located "on" (or hovers over) an object (thumbnail). (Recall timer/clock facility 290.) If not, no objects can be active, nor can any object be moving. Records (e.g., 304) of objects may be updated to reflect this fact in act 1610. The method 231'/1530 is then exited via RETURN node 1680. Returning to conditional branch point 1605, if the cursor is located on (or is hovering over) an object, the object is "active" (and therefore, other objects may be not "active".) Records (e.g., 304) of objects may be updated to reflect this fact in act 1615. Processing then continues to conditional branch point 1616.

At conditional branch point 1616, it is determined whether or not a "float" command is entered. Recall from the description of Figure 17 above that a float command may be effected by a single mouse click when a cursor is over an object which is below the surface layer of the data tank. If a float command is entered, the record of the object is updated to reflect this fact and the depth level of the object is set to the surface level as shown in act 1617, and a "click count" is updated as shown in act 1618, from which processing continues to conditional branch point 1620. Returning to conditional branch point 1616, if a "floated" command is not entered, processing continues directly to conditional branch point 1620.

At conditional branch point 1620, it is determined whether or not the "active" object is "selected". If so, then the object is "selected" (and therefore, other objects may be not "selected"). Records (e.g., 304) of objects may be updated to reflect this fact in act 1625, and a click count is incremented as shown in act 1630. The method 231'/1530 is then left via RETURN node 1680. Returning to conditional branch point 1620, if the "active" object is not "selected", processing continues to conditional branch point 1635. At conditional branch point 1635, it is determined whether or not the "active", perhaps "floated", but not "selected", object is "moving" (or being dragged). If not, processing continues to conditional branch point 1640 which determines whether or not the object just transitioned from "moving" to "not moving". That is, conditional branch point 1640 determines whether or not the object was dragged (i.e., moving) in the last cycle. If so, as shown at act 1645, the location(s) of the object(s) are stored to a persistent storage means (Recall storage device 280 of Figure 2A.), and processing continues at conditional branch point 1650. Otherwise, processing continues directly to conditional branch point 1650. At conditional branch point 1650, it is determined whether an animation is running. If so, as shown in act 1655, the animation is updated (Recall records 850.) and the process 231'/1530 is left via

return node 1680. If not, the process 231'/1530 is left directly, via RETURN node 1680.

Returning to conditional branch point 1635, if  
5 the object is being "moved", then no other objects will  
be "moving" (though they may be "pushed" as discussed  
below). Records (e.g., 304) of objects are updated to  
reflect this fact in act 1660. The object's new location  
is determined in act 1665. Recall that total object  
10 (thumbnail) occlusion may be avoided. In conditional  
branch point 1670, it is determined whether or not the  
"active" and "moved" object is within a predetermined  
distance of any other object (within the same depth  
layer, within a predetermined number of adjacent depth  
15 layers, or within any depth layer). If not, processing  
continues at conditional branch point 1650 which was  
described above. If, on the other hand, the "active" and  
"moved" object is within a predetermined distance of  
another object(s), total object (thumbnail) occlusion is  
20 avoided in act 1675 and processing continues at  
conditional branch point 1650. Exemplary processes for  
avoiding object occlusion are discussed below with  
reference to Figures 18A and 18B.

25 In each of the following exemplary object  
occlusion avoidance methods, keep in mind that three  
alternatives are possible. In the first alternative,  
only the objects in the same depth level as the moving

object are considered. In the second alternative, objects within a predetermined number of depth levels of the depth level of the moving object are considered. Finally, in the third alternative, all objects are

5 considered, regardless of their depth level. Thus, the first and third alternatives are simply special cases of the second alternative. That is, if the predetermined number of depth levels is zero (0), the first alternative results, and if the predetermined number of depth levels

10 is set to the total number of depth levels in the tank, the third alternative results. Having described these three alternatives in which each of the exemplary methods can be used, two exemplary methods for avoiding object occlusion are now described with reference to Figures 18A and 18B.

15

Figure 18A is a flow diagram of an exemplary process 1675' for avoiding object (thumbnail) occlusion assuming a "continuous with push" layout type. (Recall

20 field 338 of Figure 3.) First, at conditional branch point 1805, it is determined whether or not the "active" and "moving" object has moved (a predetermined distance) away from a stored location (or, in an alternative embodiment, the present location) of the other object.

25 If not, as shown in act 1815, the other object is "pushed" and the record (e.g., 304) of the other object is updated to reflect this fact as shown in act 1820. Further, in one embodiment, the pre-push location of the



"pushed" object is stored. (Recall, e.g., field 310 of Figure 3.) Then, as shown in act 1825, the location of the "pushed" object is updated to maintain at least a predetermined minimum distance between it and the

5 "active" and "moving" object. (Recall location field 308.) The "pushed" object may move out of the way of the "moved" object by effecting an animation, which may last about one second. Alternatively, the faster the "moved" object is moving, the faster the "pushed" objects

10 may move aside. Thus, in this embodiment, the "moved" object is like a person walking or running through tall grass (pushed objects) which move out of the person's way but resume their position after the person leaves. The method 1675' is then left via RETURN node 1830.

15           Returning to conditional branch point 1805, in this embodiment, if the "active" and "moving" object has moved away from the pre-push location (e.g., stored in field 310 of the record 304 of the "pushed" object), of

20 the "pushed" object, then the location of the "pushed" object (e.g., location field 308) is set back to its pre-push location as shown in act 1810. This may be effected with an animation which may take about one half of a second.

25           Figure 18B is a flow diagram of an alternative exemplary process 1675'' for avoiding object (thumbnail) occlusion, again assuming a "continuous with push" layout

type. (Recall field 338 of Figure 3.) First, as shown  
in act 1850, the other object is indicated as being  
"pushed". Next, in act 1855, the location of the other  
object is updated to maintain at least a minimum distance  
5 between it and the "moved" object. Then, at conditional  
branch point 1860, it is determined whether the updated  
location of the "pushed" object is within a predetermined  
distance of a location of any other objects. If so,  
those further objects must also be "pushed" to avoid  
10 total occlusion by, or to avoid totally occluding, the  
"pushed" object, and processing continues at act 1850.  
That is, the minimum distance condition is transitively  
propagated, if necessary, to displace neighboring object  
thumbnails. Otherwise, the method 1675'' is left via  
15 RETURN node 1865.

In this alternative way of avoiding object  
occlusion, the pre-push position of object thumbnails  
need not be stored. When a moving operation ceases  
20 (e.g., when the left mouse button is released during a  
drag operation), all object thumbnails maintain their  
location at that time.

Assuming that a "gridded" layout is used  
25 instead of a "continuous with push" layout, when an  
active object is released, it is anchored to the nearest  
available (i.e., having no other anchored object) grid

point. In this way, total object occlusion is avoided.  
One object may partially occlude an other(s).

Recall from Figure 2C above that a state update  
5 management process 230 may use a object depth and fade or  
shade determination process 232. Figure 19 is an  
exemplary method 232'/1540 for performing this process.  
The acts of this method may be performed for each of the  
objects. Thus, the method defines a loop through each of  
10 the objects. As shown in act 1910, an object counter is  
initialized (OB = 1). Next, as shown in conditional  
branch point 1920, it is determined whether or not the  
object is "floated". Since "floated" objects are on the  
surface of the tank, their depth need not be determined.  
15 Thus, if an object is "floated", processing branches to  
act 1940 at which the shade or fade of the object is  
determined. If it can be assumed that "floated" objects  
are not shaded or faded, processing can branch to  
conditional branch point 1950 as shown by the dashed  
20 line. Returning to conditional branch point 1920, if the  
object is not "floated", processing continues to  
act 1930. At act 1930, the depth level of the object is  
determined based on a function of one or more parameters.  
As discussed above, the function may be linear,  
25 polynomial, or exponential. Mathematically, any  
properties or parameters can be used, so long as each has  
a "partial ordering" on it. For example, the  
parameter(s) may include one or more of the following:

click history, age, time since last use, size, file type, associated application, classification, author, etc.

Next, as shown in act 1940, the shade or fade of the object is determined. At conditional branch point 1950,

5 it is determined whether there are any more objects to process. If so, processing branches to act 1960 where the object counter is incremented ( $OB = OB + 1$ ).

Processing then continues back to conditional branch point 1920. Returning to conditional branch point 1950,

10 if no more objects are to be processed, the

method 232'/1540 is left via return node 1970. Referring back to conditional branch point 1920 and acts 1930 and

1940, the fade and/or shade of a floated object could be calculated just like any other object, in which case the

15 conditional branch point 1920 would not be necessary.

The conditional branch point 1920 would be used, however, if floated objects were given special treatment (such as special highlighting for example).

20

#### **§ 4.2.3.4 MANAGING SPECIAL SERVICES**

In the following, the processes related to the management or special services are described, with

reference to Figure 2D, in § 4.2.3.4.1. Thereafter,

25 techniques for performing an implicit query function are

described in § 4.2.3.4.2 and techniques for performing a proximity clustering function are described in

§ 4.2.3.4.3.

**§ 4.2.3.4.1 PROCESSES FOR MANAGING  
SPECIAL SERVICES**

5           Having described the stored state  
information 240, as well as the processes related to the  
input management process 220, the processes related to  
the special services management process 250 are now  
described.

10           If the user interface of the present invention  
is to help a user to find objects related to an "active"  
object, the special services management process 250  
invokes the implicit query process (or more generally, an  
15 "object matcher") 252, which may use matching (e.g.,  
correlation or similarity) algorithms or heuristics,  
based on stored topic, or keywords, or contents of the  
object, or which may use predetermined object clusters,  
(See, e.g., field 704 of record 700.) stored in storage  
20 means 254, to determine objects related to the "active"  
objects. The "object matcher" can also use properties of  
the object that are not, strictly speaking, "contents of  
the object" but could be metadata such as age, size,  
usage pattern, etc. These characteristics, or properties  
25 of the object might be stored in an external fashion  
(therefore not "contents", but could still be used as the  
key by which the "object matcher" computes inter-object  
similarity. A co-occurrence method and a content-based

method, each of which may be used by the implicit query process, are briefly introduced here.

The first technique is derived from a  
5 page-to-page co-occurrence matrix based on a number  
(e.g., seven) of previous users' categorizations. Thus a  
pair of pages may match anywhere between 0 and n (e.g.,  
seven) times, where n is the number of previous users'  
categorizations. This technique essentially tells the  
10 user, "Other people thought these pages were related."  
For the content-based similarity technique, the known  
feature vector space model from information retrieval may  
be used. Objects may be pre-processed to remove the HTML  
markup. Words on a standard stop list of common words  
15 along with web-specific words may be omitted, and white  
space or punctuation may be used to delimit words.  
Phrases may also be used. Each object may be represented  
as a vector of words (and/or phrases) with entries  
representing the frequency of occurrence of a word (or a  
20 phrase) in that object. The similarity between objects  
may be measured by taking the "cosine distance" between  
the two object vectors -- that is, the dot product of the  
vectors divided by the lengths of the objects. Objects  
may be considered to be "related" if the degree to which  
25 they match exceeds a match threshold stored in field 340.  
The display may render a control object, such as a slider  
for example, for permitting the user to vary the match

threshold and see the objects found to be "related" to an active object.

Alternatively, a filtering or sorting process  
5 may be used to determine objects related based on a property, such as size, creation date, last used date, directory location, etc., of the objects. Such related objects may be visually highlighted to the user, by providing a colored halo around such objects which  
10 matches a colored halo around the "active" object for example. Alternatively, the halo of related objects may use a different color, or a different shade of the same color, to distinguish them from the "active" object.

15 The degree to which objects match, as determined by the implicit query process 252, may also be depicted. Implicit query processing and match indication may be carried out as described in U.S. Patent Application Nos. 09/152,713 and 09/152,525, entitled,  
20 "Methods, Apparatus, and Data Structures for Providing a User Interface which Exploits Spatial Memory in Three Dimensions, to Objects and which Visually Groups Matching Objects" and "Methods, Apparatus, and Data Structures for Providing a User Interface, which Exploits Spatial  
25 Memory, to Objects", respectively, both filed on September 14, 1998 (each of which is incorporated by reference).

Referring still to Figure 2D, to provide further help to the user, it may be assumed that users will position related objects in close proximity. The special services management process 250 may invoke a proximity cluster determination process (or more generally, a "proximity clustering facility") 256 to cluster object thumbnails based on proximity. Proximal clustering processing and cluster indication may be carried out as described in U.S. Patent Application Nos. 09/152,712 and 09/152,525, entitled, "Methods, Apparatus, and Data Structures for Providing a User Interface, which Exploits Spatial Memory in Three Dimensions, to Objects and which Visually Groups Proximally Located Objects" and "Methods, Apparatus, and Data Structures for Providing a User Interface which Exploits Spatial Memory, to Objects" respectively, both filed on September 14, 1998 (each of which is incorporated by reference). As was the case with the various object occlusion avoidance techniques, the proximity cluster determination process may consider only those objects at the same depth level as the active object, objects within a predetermined number of depth levels of the active object, or without regard to the depth levels of the objects.



#### **§ 4.2.3.5 MANAGING OUTPUTS**

In the following, processes which may be used to facilitate managing outputs in the user interface of the present invention are described, with reference to Figure 2E, in § 4.2.3.5.1. Then, exemplary methods for effecting such processes are described, with reference to Figure 21, in § 4.2.3.5.2.

##### **§ 4.2.3.5.1 PROCESSES FOR MANAGING OUTPUTS**

Having discussed processes related to the input management process 220, the state update management process 230, and the special services management process 250, processes related to the output management process 260 are now described below with reference to Figure 2E. Basically, the output management process 260 manages a video output process 261 and an audio output process 268. Each is described below.

The video output process (or more generally, a "video output facility") 261 uses parameters of the "data tank" 264 to render it on a video monitor 270a. Recall that the tank parameter may include a value in a tank contents field 328 of Figure 3. For each object, the video output process 261 obtains the location and depth information in fields 308 of the temporary storage means 240, and the low resolution images in fields 404 of

the object records 400 at storage means 266 to render the object thumbnails in the data tank at the appropriate position and at the appropriate depth.

5           If an object is "active", the output management process invokes the pop-up title bar generation process (or more generally, a "pop-up bar facility") 262 which accesses title (or other descriptive) information stored in fields 606 of records 600 stored in storage means 263. 10 This title (or other descriptive) information is provided to the video output process 261 so that a pop-up title bar may be rendered. A colored halo may also be rendered around an active object.

15           If an object is "selected", a preferred object viewing process (or more generally, a "preferred object viewer") 264 is invoked by the video output process 261. The preferred object viewing process 261 moves the object thumbnail associated with the "selected" object to a 20 preferred viewing location on the display. A "selected" object may be rendered in the center of display as an enlarged image. The low resolution image of the object may be replaced with the high resolution image of the object. Also recall that the thumbnail may display 25 alternative representations of the content from a given object such as schematic or iconic views. To facilitate consistency of the user interface to the user, a movement animation may be used and may last about one second.

During this enlargement, a natural transition via hardware or software supported blending may also be supported. The moved object thumbnail may, alternatively, be the actual object "live" on an associated application, such as an HTML page on an Internet browser, a text document on a word processor, a spreadsheet on an accounting application, etc. Also recall that the object thumbnail may display alternative representations of the object, such as iconic or schematic views. During a movement to the actual object, "live" in its associated application, a transition animation may employ blending or morphing.

Referring to Figure 8, each object has an animation record 850, which may be stored collectively as animation files 800. Each animation record 850 may include a field 802 for identifying the object, a field 804 for defining an animation type (e.g., morph, blend, enlarge, shrink, etc.), a field 806 for defining a start position (e.g., the object's location), a field 808 for defining an end position (e.g., the center foreground), a field 810 for defining the rate of the animation (e.g., total time = 1 second), and a field 812 for tracking the percent complete (i.e., a current state) of the animation.

Finally, the video output process 261 may invoke a perspective view and shadow determination

process 265 to render the object thumbnails to simulate a perspective view in three-dimensions such that the perceived sizes of the thumbnails are different since they are rendered from the perspective of the viewing point and to cast a simulated shadow on the floor 912 of the data tank.

Having described the processes associated with the video output process 261, a process associated with an audio output process 268 is now described. Still referring now to Figure 2E, an audio cue process (or more generally, an "audio cue player") 269 which may be used to render stored audio data (not shown) based on the state(s) of object(s). For example, when an object is "activated", an associated audio cue may be played. This may be an abstract percussive "pong" sound. When an object is "floated", an associated audio cue may be played. This may be a gentle "bubbling" sound. When an object is "selected", the movement animation may be accompanied by an associated audio cue. This may be an abstract "whoosh" sound of increasing pitch and loudness in which the parameters of a flanger effect are changed over the duration of the sound. When a "selected" object is deselected, an animation, moving the object away from the user so that it appears smaller, may be accompanied by an associated audio cue. This may be a reversed version of the "whoosh" sound. When an object is being "moved", an associated audio cue may be played. This may

be an oscillating electronic hum. The pitch of the oscillating hum may vary with the speed at which the object thumbnail is being moved. Left/right (or up/down) balance of the audio may be varied based on the relative position of the object on the screen. When an object is initially "moving", a mechanical latching sound cue may be played. When the object transitions out of a "moving" state, a mechanical unlatching sound cue may be played. When an object is being "pushed", an associated audio cue (e.g., based on a shuffling card sound) may be played.

Although the processes discussed above were described as being associated with input, state update, special services, and output management processes, as is apparent to those skilled in the art, certain of the operations can be combined or can be separated out.

#### **§ 4.2.3.5.2      EXEMPLARY METHODS FOR MANAGING OUTPUTS**

Recall from Figure 13 that the animation loop process 1300 includes a act 260'/1350 of rendering a display and generating audio based on updated states. Figure 21 is a flow diagram of an exemplary method 260' for performing an output management process for generating an audio and video output. First, as shown in act 2105, the cursor is displayed at its location. (Recall cursor location determination process 237, cursor location field 324, and video output process 261.) Next,

as shown in act 2110, the data tank is rendered. (Recall storage means 264, video output process 261, and display 270a.) Next, a loop in which all objects are processed is run.

5

The object loop is initialized at act 2115 by initializing an object count value (OB\_CT = 1). Next, in act 2120, the object (thumbnail) is displayed anchored at its location, at its depth level. Processing continues at conditional branch point 2125 where it is determined whether or not the object being processed is "active". (Recall field 312.)

10

15

20

25

If the object is not active, processing branches to conditional branch point 2130 where it is determined whether the object being processed is "selected". (Recall field 314.) If so, a high resolution version of the moved object thumbnail is ultimately displayed at the preferred viewing location, as shown in act 2135, by progressing through the animation associated with the object (unless the animation is 100 percent complete). Recall that, in an alternative embodiment, a "live" object may be loaded into an associated application and then displayed. If the object being processed underwent a "not selected" to "selected" transition, the movement animation and associated audio cue may be played. To determine transitions, the object's previous state may be used.

From act 2135, or directly from conditional branch point 2130 if the object being processed is not "selected", the object indexing count is incremented (OB\_CT = OC\_CT + 1) at act 2140. Then, as shown in act 2145, if all of the objects have been processed (OB\_CT > MAXIMUM OB\_CT; Recall field 326.), then the process 260' is exited via return node 2195. If, on the other hand, more objects need to be processed, processing branches back, via node A 2150, to act 2120.

Returning to conditional branch point 2125, if the object being processed is "active", then processing branches to act 2155 in which a pop-up title bar is displayed over the object (thumbnail). Next, at conditional branch point 2160, it is determined whether the object being processed just transitioned to the "active" state from an inactive state. Recall that to determine transitions, the object's previous state may be used. If so, an activation audio cue is played as shown in act 2165. From act 2165, or from conditional branch point 2160 if the object was already "active", act 2170 is performed. In act 2170, a colored halo is displayed around the periphery of the object (thumbnail). Next, in act 2175 if other objects are found to match the active object to a predetermined degree (Recall match threshold 340.), visual indications of such a match (or

non-match) are rendered. (Recall implicit query process 252.)

Next, as shown at conditional branch point 2180, if the object being processed is "moving" (Recall field 316.), associated audio is played as shown in act 2185. Recall also that transitions to and from the "moving" state may be accompanied by mechanical latching and unlatching, respectively, audio cues. From act 2185, or from conditional branch point 2180 if the object being processed is not "moving", processing continues via node C 2190 to act 2130 where processing is carried out as discussed above.

#### 4.2.3.6 STRUCTURE FOR PERFORMING PROCESSES

In the context of the personal computing environment 100 of Figure 1A, the processes discussed with reference to Figures 2A through 2E may be carried out by the following facilities. The pointer input process 214 may be performed, at least in part, by the pointer 142, the serial port interface 146 and program modules 137. All data may be stored on any of the storage or memory devices or may be received from a remote source. The video output process 270 may be performed, at least in part, by program modules 137, the video adapter 148, the graphics accelerator, and the



video monitor 147. The audio output process 278 may be performed, at least in part, by program modules 137, the sound card 161 and the speaker(s) 162. All other process may be effected by program modules 137 and application  
5 programs 136 (which may be written in C++) executed on the processing units 121.

### **§ 4.3 SUMMARY**

10 As illustrated in the foregoing description,  
the present invention provides a user interface, and in particular a graphical user interface, to information or content (also referred to as an "object"). The present invention permits a user to view and organize all objects  
15 and to edit or otherwise work on a selected object by, for example, representing, graphically, objects or content with object thumbnail which can be added, moved, or deleted from a display rendered on a video monitor. The present invention may map one or more parameters of  
20 an object to a coordinate in a direction orthogonal to the surface. The present invention may use pop-up title bars for permitting a user to discern more about the object represented by the object thumbnail. The present invention may use higher resolution image representations  
25 of objects, or "live" objects loaded within an associated application, for editing or otherwise working on a selected object.

The user interface of the present invention exploits spatial memory by, for example, providing two-dimensional layers on which the object thumbnails may be manipulated.

5

The user interface of the present invention is intuitive and minimizes the time needed for a user to become familiar with it. The user interface of the present invention is neither cumbersome, nor does it require too many different types of inputs to manipulate the object thumbnails. For example, the present invention may use inputs from a familiar input device such as a mouse or pointer to manipulate the object thumbnails. The present invention may also prevent one object thumbnail from totally occluding another so that neither is (totally) hidden from the user.

10

15

20

25

Finally, the user interface of the present invention can provide intelligent help to the user. For example, the present invention may cluster, by rendering a visual boundary for example, object thumbnails which are arranged, by the user, relatively close to one another. Further, the present invention may employ a matching (e.g., correlation or similarity) algorithm to determine whether certain objects are related. Alternatively, objects may be related by an explicit selection or designation by a user. If so, the present invention may employ some type of visual indicator(s) of

"matching" (or non-matching) objects. This visual indication may be rendered upon an event, such as when an object thumbnail is activated.

5                   Thus, the present invention represents an advance in the area of user interfaces.

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
222

WHAT IS CLAIMED IS:

1 1. A man-machine interface method for permitting a user  
2 to act on thumbnails, each thumbnail representing an  
3 associated object containing information, for use with a  
4 machine having a video display device and a user input  
5 device, the man-machine interface method comprising:

6 a) generating a three-dimensional environment,  
7 having a depth, to be rendered on the video display  
8 device;

9 b) determining a two-dimensional location and a  
10 depth of each of the thumbnails in the  
11 three-dimensional environment, wherein, for each of  
12 the thumbnails, the depth is a function of at least  
13 one parameter of the object associated with the  
14 thumbnail; and

15 c) generating the thumbnails within the  
16 three-dimensional environment, at the determined  
17 two-dimensional locations and depths, to be rendered  
18 on the video display device.

1 2. The man-machine interface method of claim 1 wherein,  
2 for each of the thumbnails, the depth is a linear  
3 function of at least one parameter of the object  
4 associated with the thumbnail.

1 3. The man-machine interface method of claim 1 wherein,  
2 for each of the thumbnails, the depth is a polynomial

function of at least one parameter of the object  
associated with the thumbnail.

4. The man-machine interface method of claim 1 wherein,  
for each of the thumbnails, the depth is an exponential  
function of at least one parameter of the object  
associated with the thumbnail.

5. The method of claim 1 wherein the at least one  
parameter includes at least one parameter selected from a  
group of parameters consisting of (a) click history, (b)  
age, (c) time since last use, (d) size, (e) file type,  
(f) associated application, (g) classification, and (h)  
author.

6. The man-machine interface method of claim 1 further  
comprising:

- d) accepting inputs from the user input device;
- e) determining a two-dimensional cursor location  
based on the accepted inputs; and
- f) generating a cursor at the determined  
two-dimensional cursor location, to be rendered on  
the video display device.

7. The man-machine interface method of claim 6 further  
comprising:

3 g) if the two-dimensional location of the cursor is  
4 located on or over one of the thumbnails, defining a  
5 state of that thumbnail as active.

1 8. The man-machine interface method of claim 7 further  
2 comprising:

3 h) generating a pop-up information bar located over  
4 the active thumbnail, to be rendered on the video  
5 display device.

1 9. The man-machine interface method of claim 7 further  
2 comprising:

3 h) if the user input provides a selection input and  
4 if an active or floated thumbnail exists, then  
5 generating a higher resolution visual representation  
6 of the object represented by and associated with the  
7 active or floated thumbnail, at a preferred viewing  
8 location at a foreground of the three dimensional  
9 environment, to be rendered on the video display  
10 device.

1 10. The man-machine interface method of claim 7 further  
2 comprising:

3 h) if the user input provides a float input and if  
4 an active thumbnail exists, then setting the depth  
5 of the active thumbnail to a predetermined value and  
6 defining a state of the active thumbnail as floated.

1 11. The man-machine interface method of claim 9 wherein  
2 the act of generating the higher resolution visual  
3 representation of the object represented by and  
4 associated with the active thumbnail includes:

5 - generating an animation which moves the  
6 higher resolution visual representation of the  
7 object represented by and associated with the  
8 active thumbnail from the location of the  
9 active thumbnail to the preferred viewing  
10 location at the foreground of the three  
11 dimensional environment, to be rendered on the  
12 video display device.

1 12. The man-machine interface method of claim 11 further  
2 comprising:

3 i) if the user input provides a deselection input  
4 and if a selected thumbnail exists, then generating  
5 a video output for moving the high resolution visual  
6 representation of the object represented by and  
7 associated with the active thumbnail to the  
8 two-dimensional location of the selected thumbnail,  
9 to be rendered on the video display device.

1 13. The man-machine interface method of claim 9 further  
2 comprising:

3 i) if the user input provides a sink input and if a  
4 floated thumbnail exists, then setting the depth of

5 the floated thumbnail to a previous value and  
6 defining a state of the floated thumbnail as active.

1 14. The man-machine interface method of claim 7 further  
2 comprising:

3 h) if the user input provides a selection input and  
4 if an active thumbnail exists, then

5 i) invoking an application related to the  
6 object represented by and associated with the  
7 active thumbnail,

8 ii) loading the object represented by and  
9 associated with the active thumbnail into the  
10 application, and

11 iii) generating a video output of the  
12 application with the loaded object represented  
13 by and associated with the active thumbnail at  
14 a preferred viewing location, to be rendered on  
15 the video display device.

1 15. The man-machine interface method of claim 9 further  
2 comprising:

3 h) if the user input provides a selection input and  
4 if a floated thumbnail exists, then

5 i) invoking an application related to the  
6 object represented by and associated with the  
7 floated thumbnail,



8           ii) loading the object represented by and  
9           associated with the floated thumbnail into the  
10          application, and  
11          iii) generating a video output of the  
12          application with the loaded object represented  
13          by and associated with the floated thumbnail at  
14          a preferred viewing location, to be rendered on  
15          the video display device.

1          16. The man-machine interface method of claim 7 further  
2          comprising:

3           h) if the user input provides a move input and if  
4           an active or floated thumbnail exists, then  
5           i) updating the two-dimensional location of  
6           the active or floated thumbnail based on the  
7           move input.

1          17. The man-machine interface method of claim 16 wherein  
2          the move input is a left button mouse drag.

1          18. The man-machine interface method of claim 1 wherein  
2          the three-dimensional environment defines a foreground  
3          and a background, and  
4          wherein the act of generating thumbnails, within the  
5          three-dimensional environment, at the determined  
6          two-dimensional locations and depths, to be rendered on  
7          the video display device, includes:

8 i) using perspective views so that any thumbnails  
9 in the foreground defined by the three-dimensional  
10 environment appear larger than any thumbnails in the  
11 background defined by the three-dimensional surface.

1 19. The man-machine interface method of claim 18 wherein  
2 a thumbnail partially occludes any thumbnails behind it,  
3 based on a viewing point.

1 20. The man-machine interface method of claim 1 further  
2 comprising:

- 3 d) accepting inputs from the user input device;  
4 e) determining a viewing point two-dimensional  
5 location, depth and direction based on the accepted  
6 inputs; and  
7 f) generating only that portion of the  
8 three-dimensional environment and only those  
9 thumbnails that are in front of the virtual viewing  
10 point determined in act (e), to be rendered on the  
11 video display device.

1 21. The man-machine interface method of claim 1 wherein  
2 the thumbnails are low resolution bit maps.

1 22. The man-machine interface method of claim 21 wherein  
2 the low resolution bit maps are 64 pixels by 64 pixels  
3 and have 24 bit color.

1 23. The method of claim 20 wherein if the depth of the  
2 viewing point is below a predetermined depth, further  
3 performing a step of:

4 g) gradually decreasing the depth of the viewing  
5 point to float the viewing point while no user  
6 inputs are received.

1 24. The method of claim 1 further comprising, for each  
2 of the thumbnails, determining a shade to be applied to  
3 the thumbnail based on its depth.

1 25. The method of claim 24 wherein the shade to be  
2 applied to the thumbnail darkens as the depth increases.

1 26. The method of claim 24 wherein the shade to be  
2 applied to the thumbnail darkens as a distance between  
3 the depth of the thumbnail and a viewing point increases.

1 27. The method of claim 1 further comprising, for each  
2 of the thumbnails, determining a fade to be applied to  
3 the thumbnail based on its depth.

1 28. The method of claim 27 wherein the fade to be  
2 applied to the thumbnail increases as the depth  
3 increases.

1 29. The method of claim 27 wherein the fade to be  
2 applied to the thumbnail increases as a distance between  
3 the depth of the thumbnail and a viewing point increases.

1 30. The method of claim 1 further comprising, for each  
2 of the thumbnails, determining a tint to be applied to  
3 the thumbnail based on its depth.

1 31. The method of claim 30 wherein the tint to be  
2 applied to the thumbnail increases as the depth  
3 increases.

1 32. The method of claim 30 wherein the tint to be  
2 applied to the thumbnail increases as a distance between  
3 the depth of the thumbnail and a viewing point increases.

1 33. The method of claim 1 wherein the three dimensional  
2 environment includes a floor, the method further  
3 comprising a step of generating a shadow, for each of the  
4 thumbnails, on the floor.

1 34. A system which permits a user to interact with  
2 thumbnails, each thumbnail representing an associated  
3 object containing information, the system comprising:  
4 a) an input facility for accepting user inputs;  
5 b) a storage facility containing  
6 i) a two-dimensional location, a depth and  
7 state information for each of the thumbnails,

- 8           ii) a two-dimensional cursor location, and
- 9           iii) a three-dimensional environment having a
- 10          simulated depth;
- 11       c) a processing unit which
- 12           i) accepts user inputs from the input
- 13           facility,
- 14           ii) updates (a) the two-dimensional location,
- 15           and state information for each of the
- 16           thumbnails contained in the storage facility,
- 17           and (b) the two-dimensional cursor location
- 18           contained in the storage facility, based on the
- 19           accepted user inputs,
- 20           iii) updates depth information for each of the
- 21           thumbnails contained in the storage facility
- 22           based on at least one parameter of the object
- 23           associated with the thumbnail, and
- 24           iv) generates video outputs based on
- 25                A) the two-dimensional location, depth
- 26                and state information for each of the
- 27                thumbnails,
- 28                B) the two-dimensional cursor location,
- 29                and
- 30                C) the three-dimensional environment,
- 31           contained in the storage facility; and
- 32       d) a video display unit for rendering the video
- 33          outputs generated by the processing unit.

1 35. The system of claim 34 wherein the state information  
2 for each of the thumbnails contained in the storage  
3 facility includes an indication of whether or not the  
4 thumbnail is active, and

5 wherein the processing unit determines that a  
6 thumbnail is active if a cursor is located on or over a  
7 thumbnail based on the two-dimensional location of the  
8 cursor and the two dimensional location of the thumbnail.

1 36. The system of claim 35 wherein the storage facility  
2 further contains descriptive textual information for each  
3 of the thumbnails, and

4 wherein, if a thumbnail is active,

5 i) the processing unit generates a pop-up bar,  
6 based on descriptive textual information, for  
7 the active thumbnail, and

8 ii) the video display unit renders the pop-up  
9 bar over the rendered t.

1 37. The system of claim 35 wherein, if a thumbnail is  
2 active or floated and the input facility accepts a  
3 selection input, then

4 i) the processing unit updates the state of  
5 the thumbnail,

6 ii) the processing unit gets a second, higher  
7 resolution, visual representation of the object  
8 represented by and associated with the  
9 thumbnail,

10           iii) the processing unit generates a video  
11           output based on the higher resolution, visual  
12           representation of the object represented by and  
13           associated with the thumbnail at a preferred  
14           viewing location, and  
15           iv) the video display device renders the video  
16           output generated by the processing unit.

1       38. The system of claim 37 further comprising an audio  
2       output device,

3           wherein the storage facility further contains a  
4       first audio cue, and

5           wherein, when an object is selected, the processing  
6       unit provides the first audio cue to the audio output  
7       device.

1       39. The system of claim 37 wherein each thumbnail is a  
2       64 pixel by 64 pixel bit map having 24 bit color and  
3       wherein each higher resolution, visual representation of  
4       the objects is a 512 pixel by 512 pixel bit map having 24  
5       bit color.

1       40. The system of claim 37 wherein the processing unit  
2       further effects a video output based on an animation of  
3       the higher resolution, visual representation of the  
4       object represented by and associated with the thumbnail,  
5       moving from the location of the thumbnail to a location  
6       at the foreground of the three-dimensional environment.

1 41. The system of claim 35 wherein if the input facility  
2 provides a float input and an active thumbnail exists,  
3 then the processing unit will set the depth of the active  
4 thumbnail to a predetermined value and will define the  
5 state of the active thumbnail as floated.

1 42. The system of claim 35 wherein, if a thumbnail is  
2 active and the input facility accepts a selection input,  
3 then

- 4 i) the processing unit updates the state of  
5 the thumbnail to selected,  
6 ii) the processing unit opens an application  
7 with which the object, associated with and  
8 represented by the selected thumbnail, is  
9 associated,  
10 iii) the processing unit loads the object into  
11 the application,  
12 iv) the processing unit generates a video  
13 output based on the object loaded onto the  
14 opened application and a preferred viewing  
15 location, and  
16 v) the video display device renders the video  
17 output generated by the processing unit.

1 43. The system of claim 41 wherein if the input facility  
2 provides a sink input and if a floated thumbnail exists,  
3 then the processing unit will set the depth of the



4 floated thumbnail to a previous value and will define a  
5 state of the floated thumbnail as active.

1 44. The system of claim 37 wherein, if a thumbnail is  
2 active or floated and the input facility accepts a move  
3 input, then

- 4 i) the processing unit updates the state and  
5 location of the thumbnail,
- 6 ii) the processing unit generates a video  
7 output based on the updated location of the  
8 thumbnail, and
- 9 iii) the video display device renders the  
10 video output generated by the processing unit.

1 45. The system of claim 35 wherein if a thumbnail is  
2 floated and the input facility accepts a selection input,  
3 then

- 4 i) the processing unit updates the state of the  
5 thumbnail to selected,
- 6 ii) the processing unit opens an application with  
7 which the object, associated with and represented by  
8 the selected thumbnail, is associated,
- 9 iii) the processing unit loads the object into the  
10 application,
- 11 iv) the processing unit generates a video output  
12 based on the object loaded onto the opened  
13 application and a preferred viewing location, and

14           v) the video display device renders the video  
15           output generated by the processing unit.

1       46. The system of claim 34 wherein the storage facility  
2       further contains virtual viewing point location  
3       information,

4           wherein the input facility includes a mouse, and  
5           wherein the processing unit

6           d) accepts inputs from the user input device;

7           e) determines a viewing point location and  
8           direction based on the accepted inputs; and

9           f) generates only that portion of the  
10          three-dimensional environment and only those  
11          thumbnails that are in front of the virtual viewing  
12          point determined in step (e), to be rendered on the  
13          video display device.

1       47. A machine readable medium containing data and  
2       machine executable instructions which, when executed by a  
3       machine, performs the method of claim 1.

1       48. A machine readable medium containing data and  
2       machine executable instructions which, when executed by a  
3       machine, performs the method of claim 6.

1       49. A machine readable medium containing data and  
2       machine executable instructions which, when executed by a  
3       machine, performs the method of claim 7.

1 50. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 8.

1 51. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 9.

1 52. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 10.

1 53. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 11.

1 54. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 12.

1 55. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 13.

1 56. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 14.

1 57. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 15.

1 58. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 18.

1 59. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 19.

1 60. A machine readable medium containing data and  
2 machine executable instructions which, when executed by a  
3 machine, performs the method of claim 20.

ABSTRACT

A graphical user interface in which object thumbnails are rendered in a three-dimensional environment and which exploits spatial memory. The objects may be moved, continuously, with a two-dimensional input device.

- 5 Pop-up title bars may be rendered over active objects. Intelligent help may be provided to the user, as visual indicators, based on proximity clustering or based on matching algorithms. The simulated location of the object thumbnails in a direction orthogonal to the surface is
- 10 based on function, such as a linear, polynomial, or exponential function for example, of one or more object properties, such as number of mouse clicks since selected, age, size, etc.

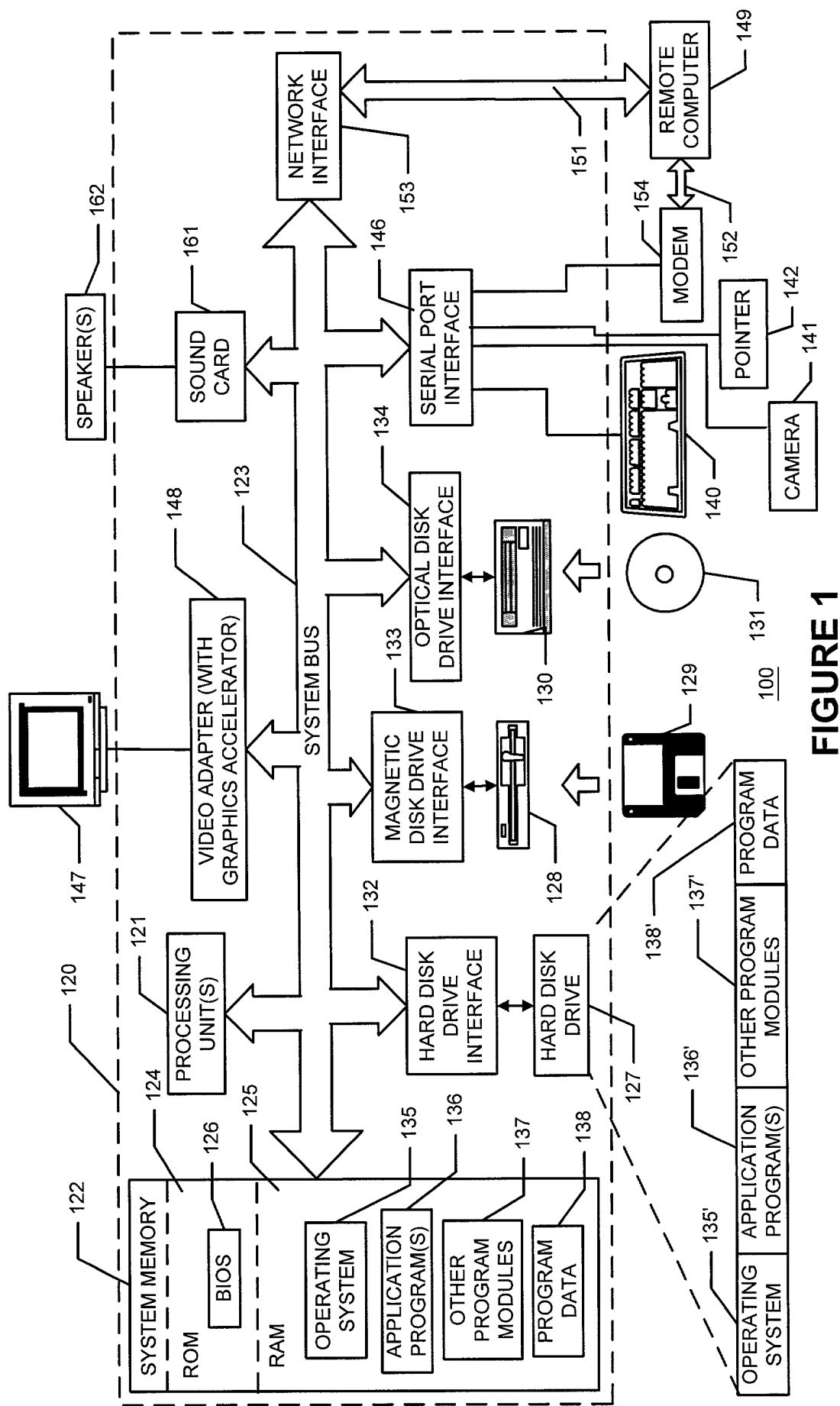
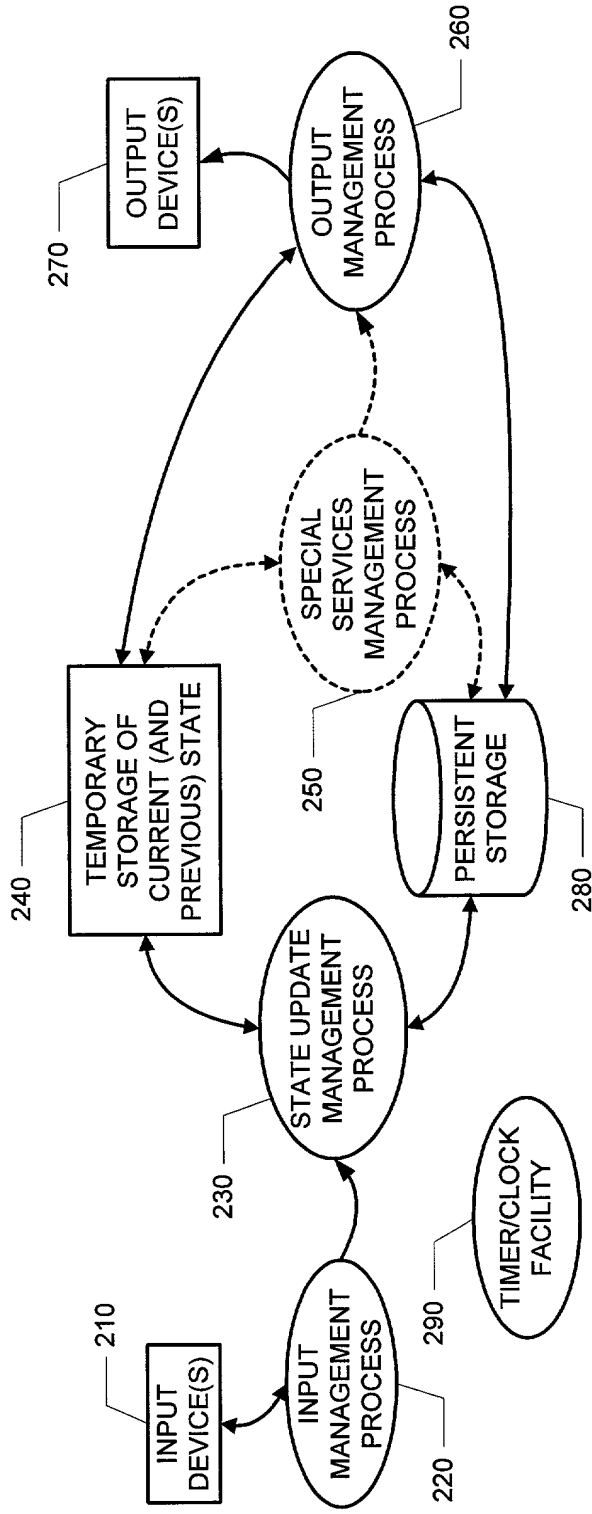


FIGURE 1



200

FIGURE 2A

FIGURE 2B

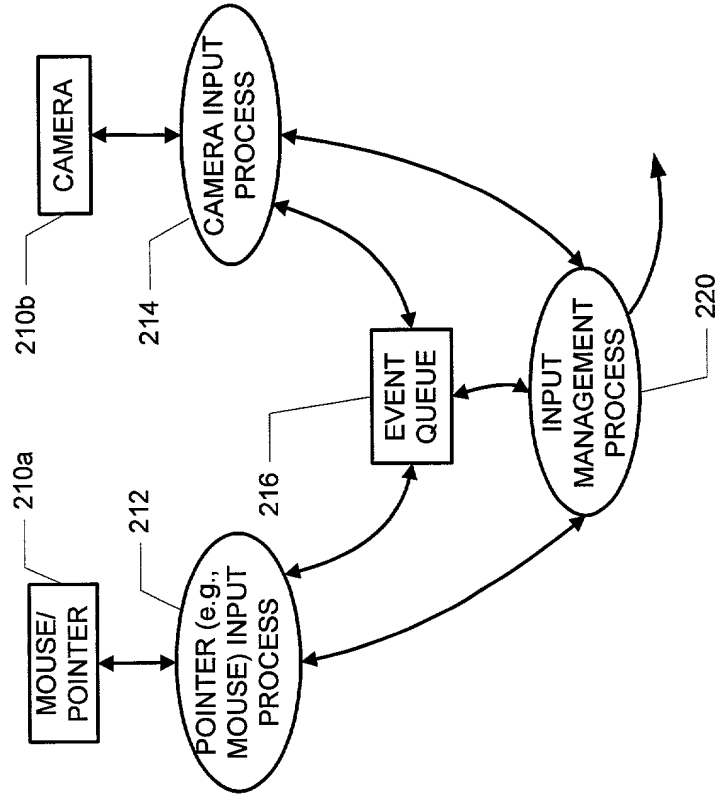


FIGURE 2C

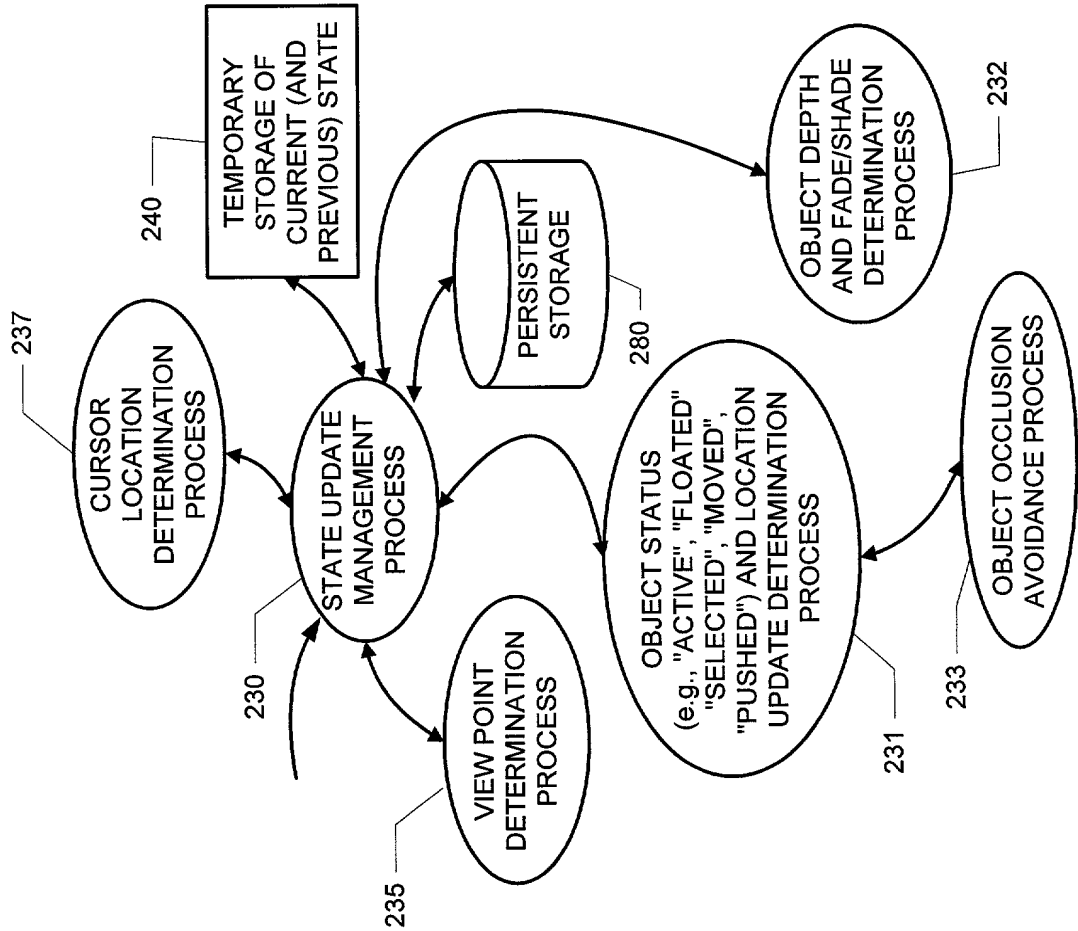




FIGURE 2D

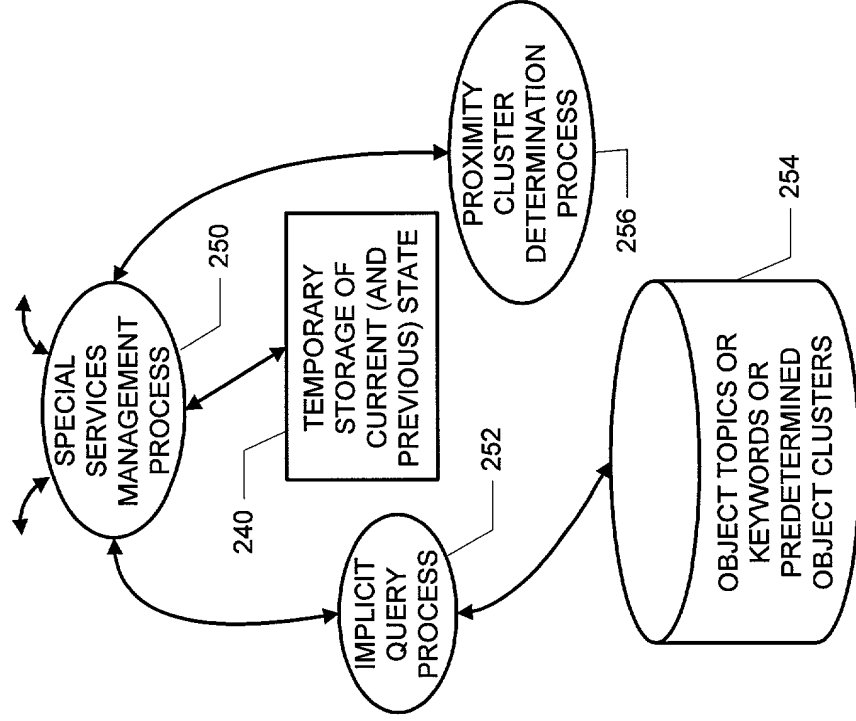
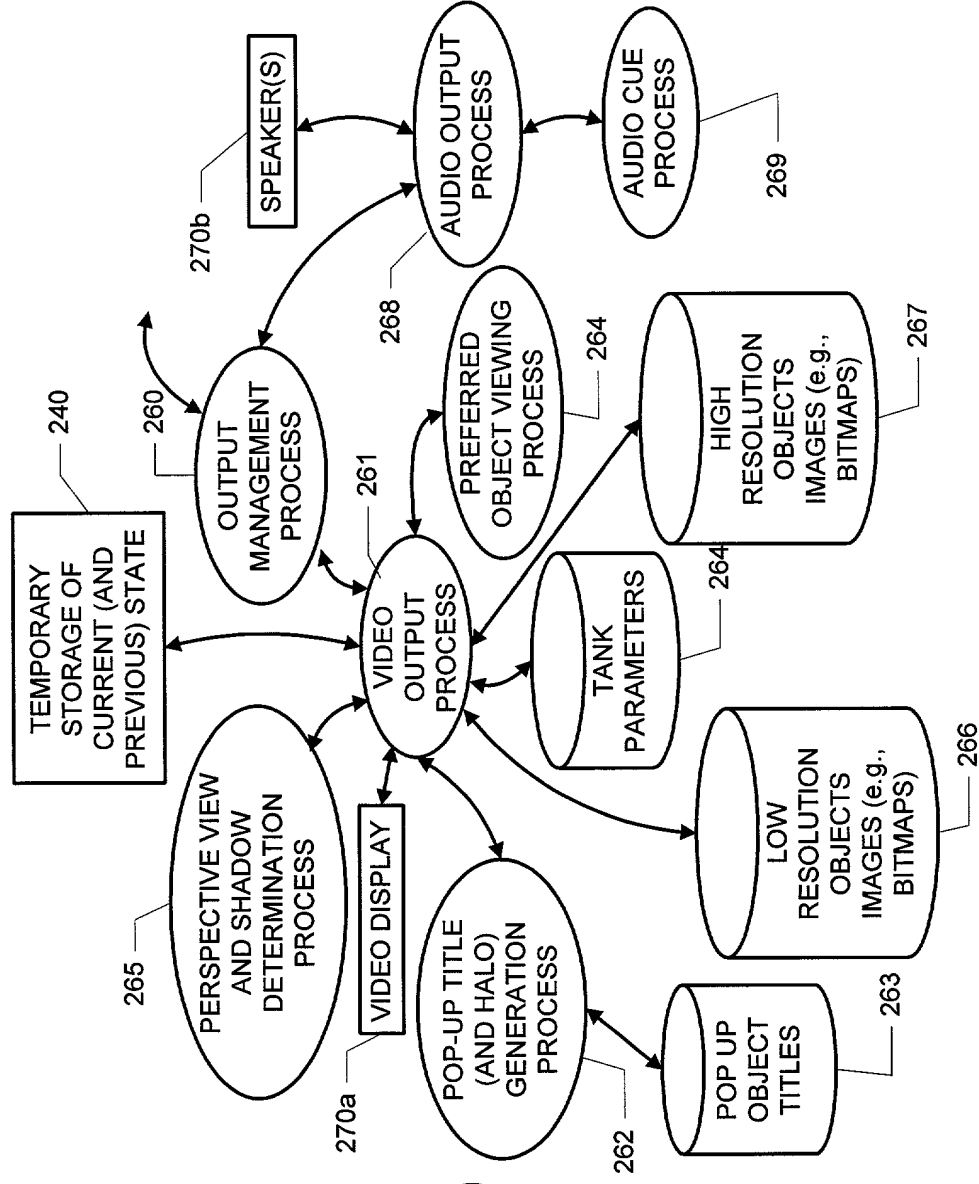
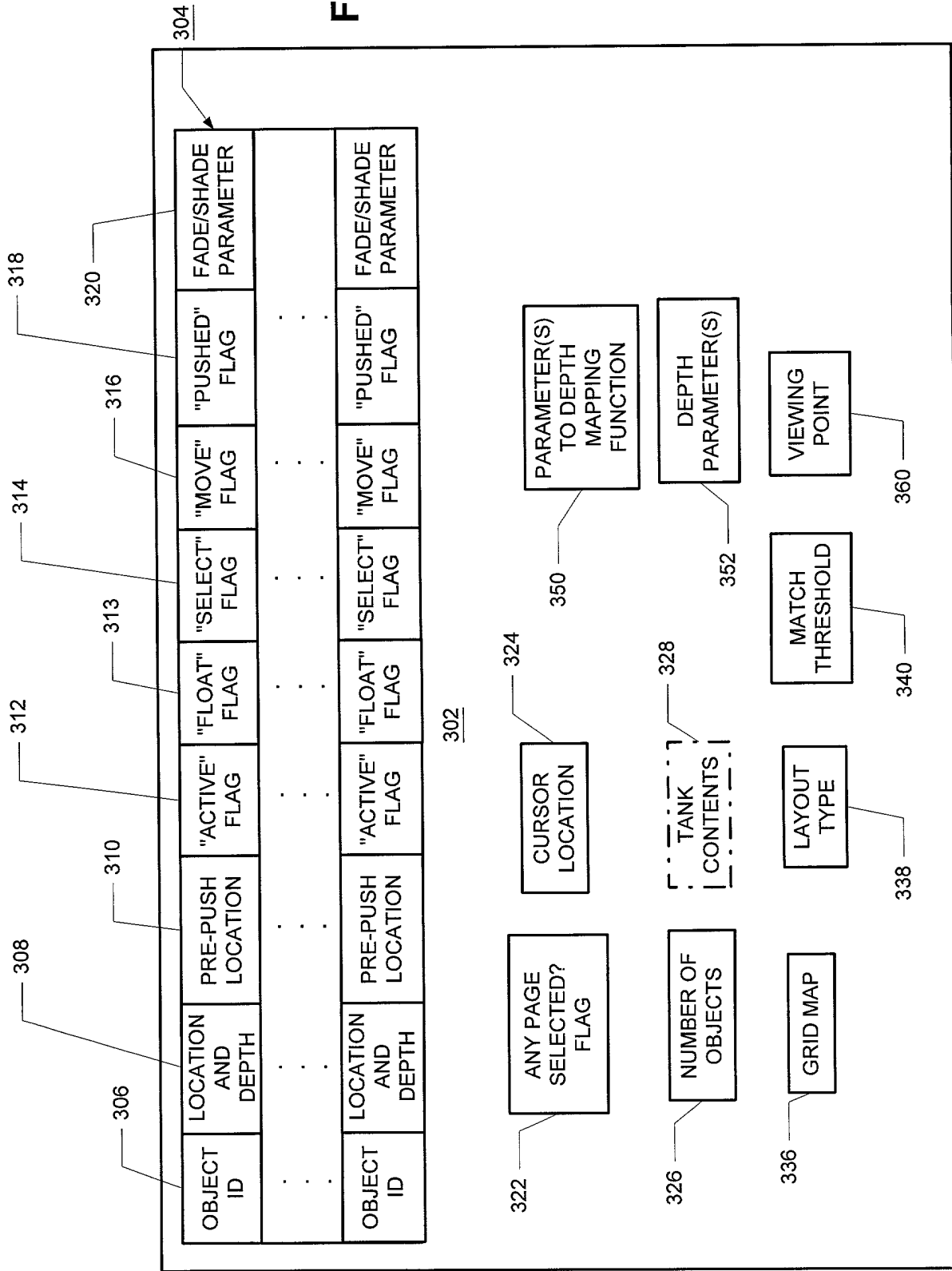


FIGURE 2E





**FIGURE 3**  
300

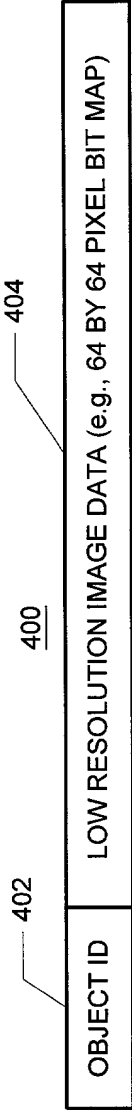


FIGURE 4

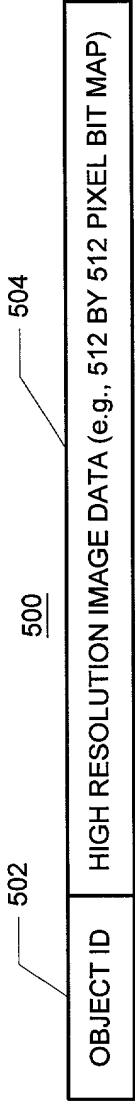


FIGURE 5

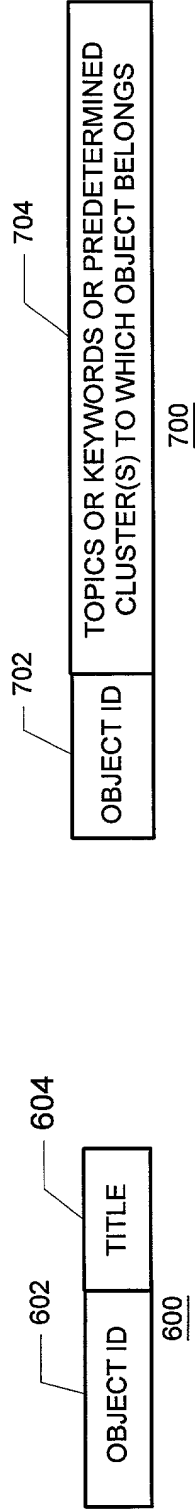


FIGURE 6

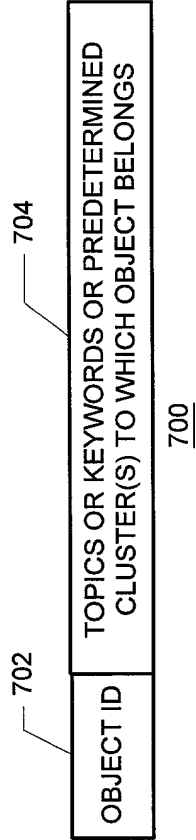


FIGURE 7

OBJECT ID	ANIMATION TYPE	START POSITION	END POSITION	RATE	PERCENT COMPLETE
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:

800

FIGURE 8

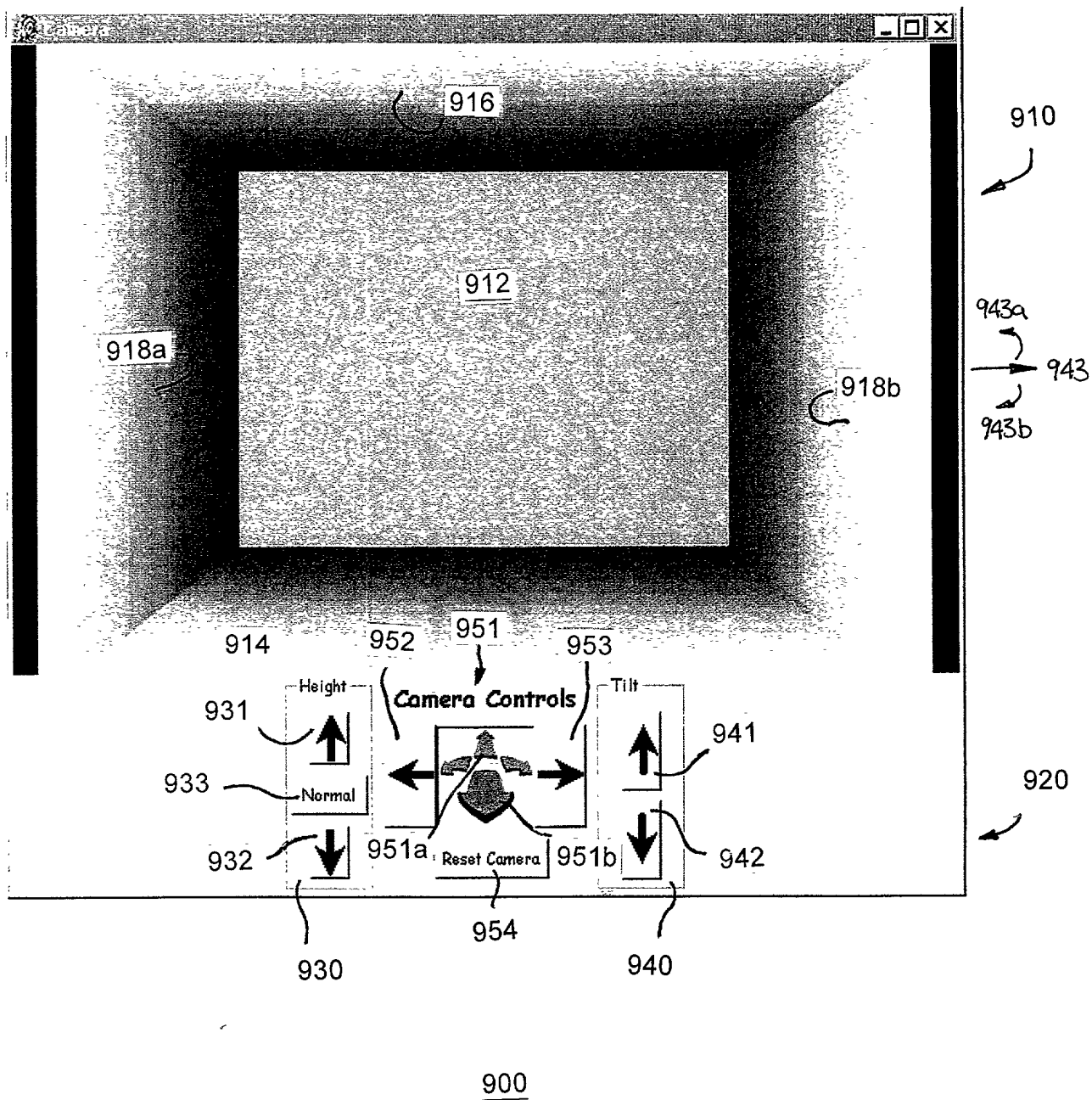
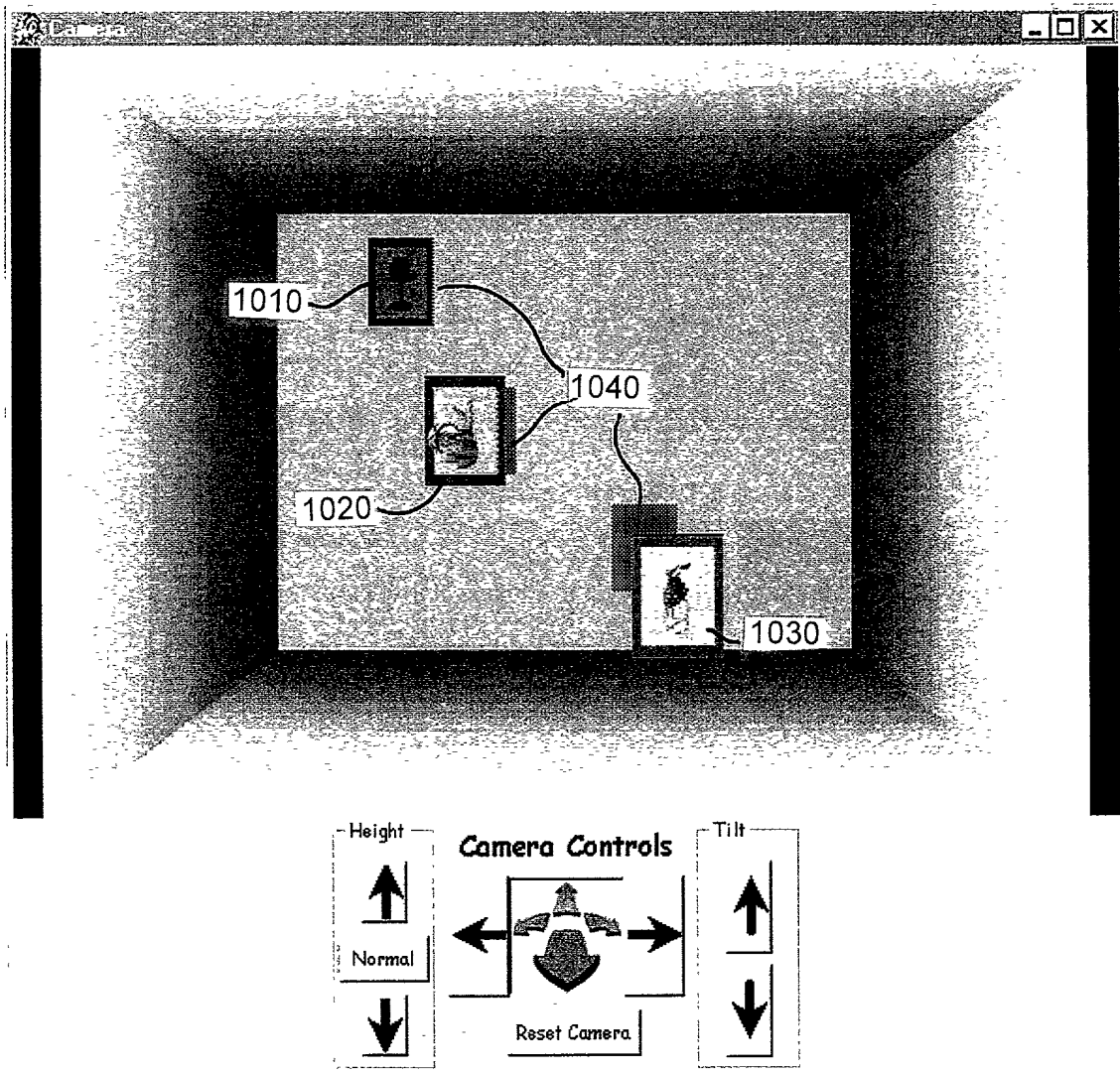


FIGURE 9

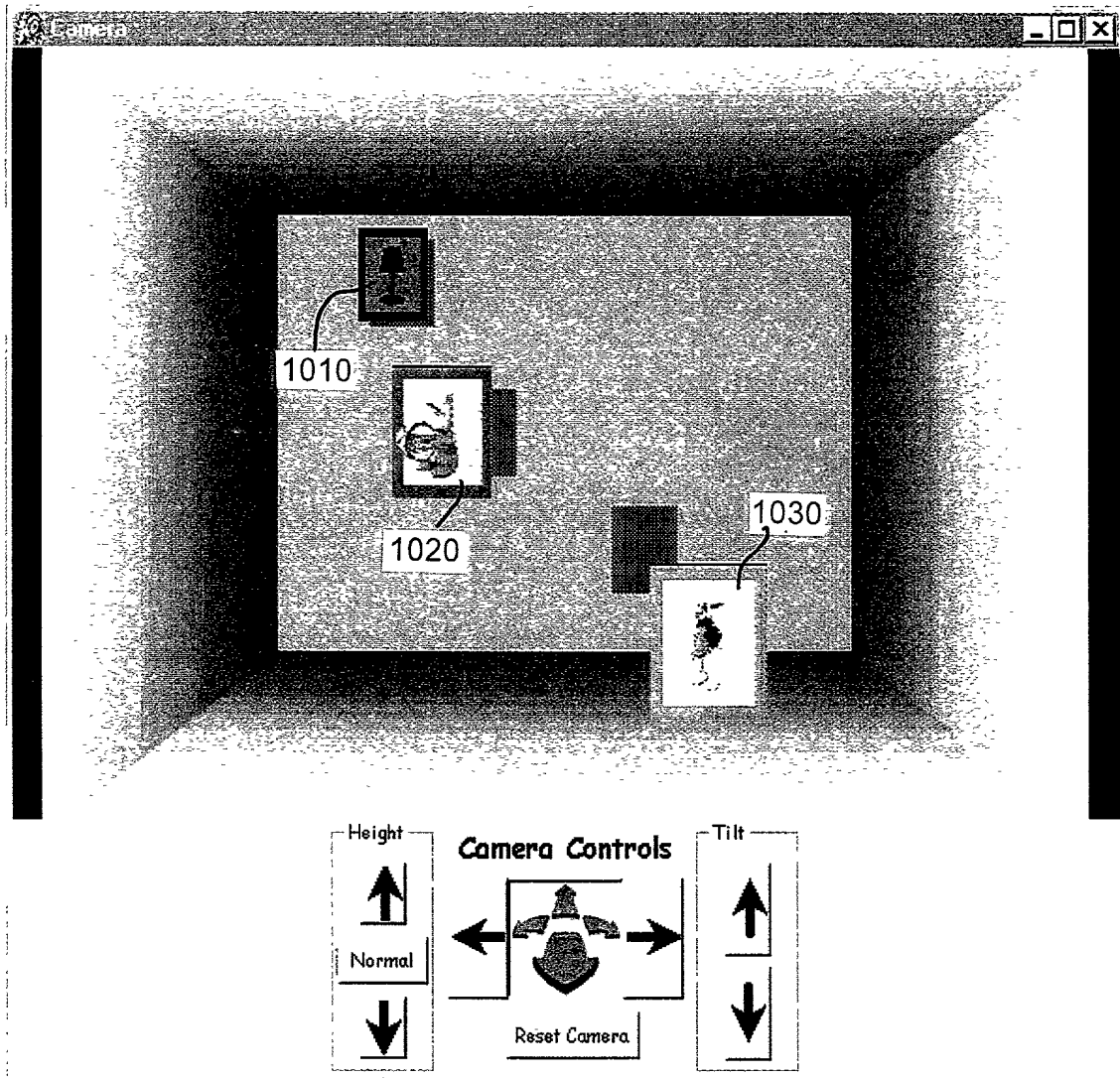


1000

**FIGURE 10A**

The screenshot shows a software window titled "Camera". Inside the window is a 3D perspective view of a scene. The scene contains three objects: a small dark cube labeled "1010" at the top left, a larger white rectangular block labeled "1020" in the center with a wireframe model on its front face, and a smaller white rectangular block labeled "1030" at the bottom right. Below the 3D view is a "Camera Controls" panel. This panel has two vertical sliders: "Height" on the left and "Tilt" on the right. Each slider has an upward-pointing arrow at the top and a downward-pointing arrow at the bottom. Between the sliders is a central area with a camera icon and four arrows pointing up, down, left, and right. Below this central area is a button labeled "Reset Camera".

**FIGURE 10B**



1000"

**FIGURE 10C**



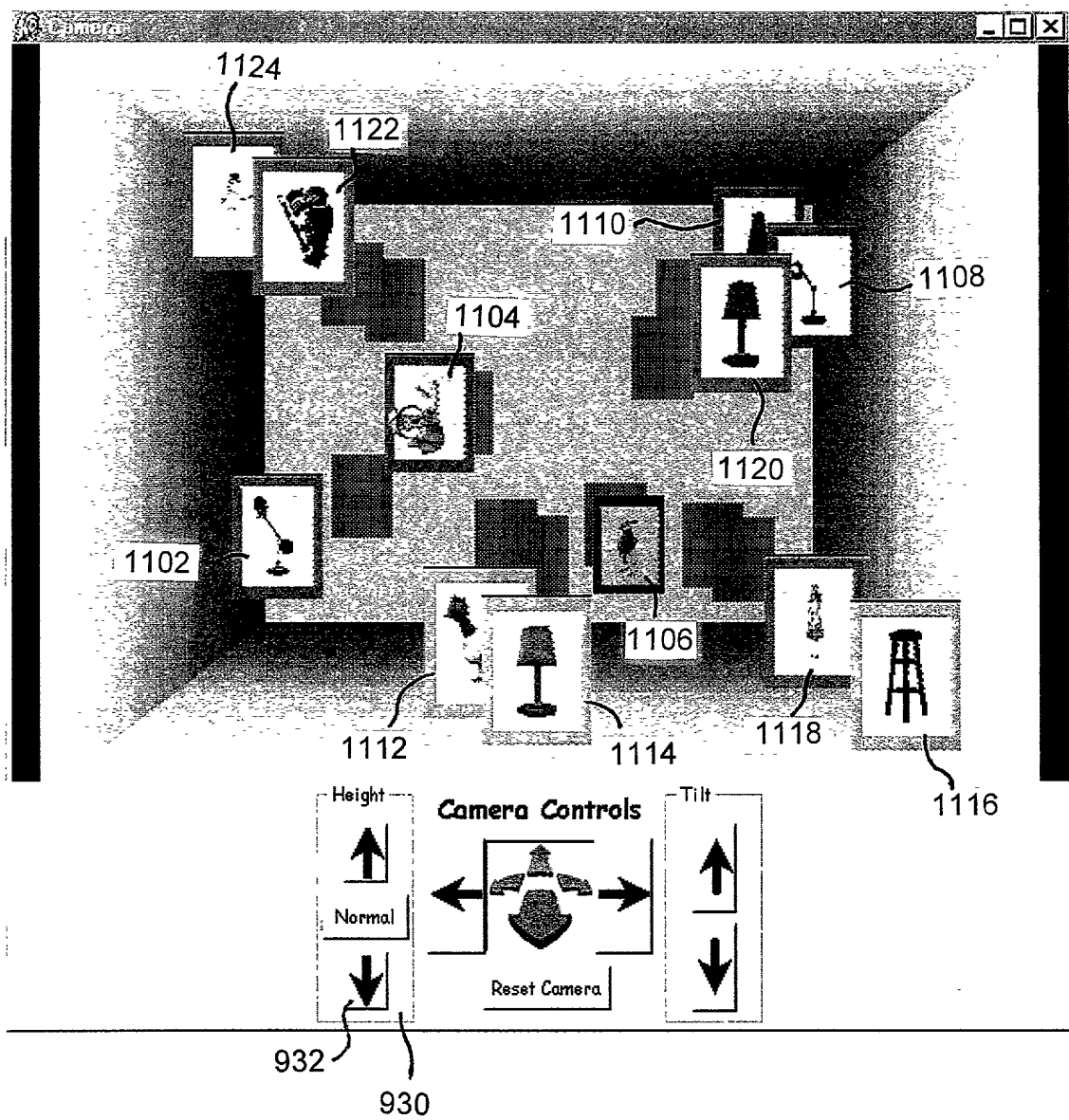
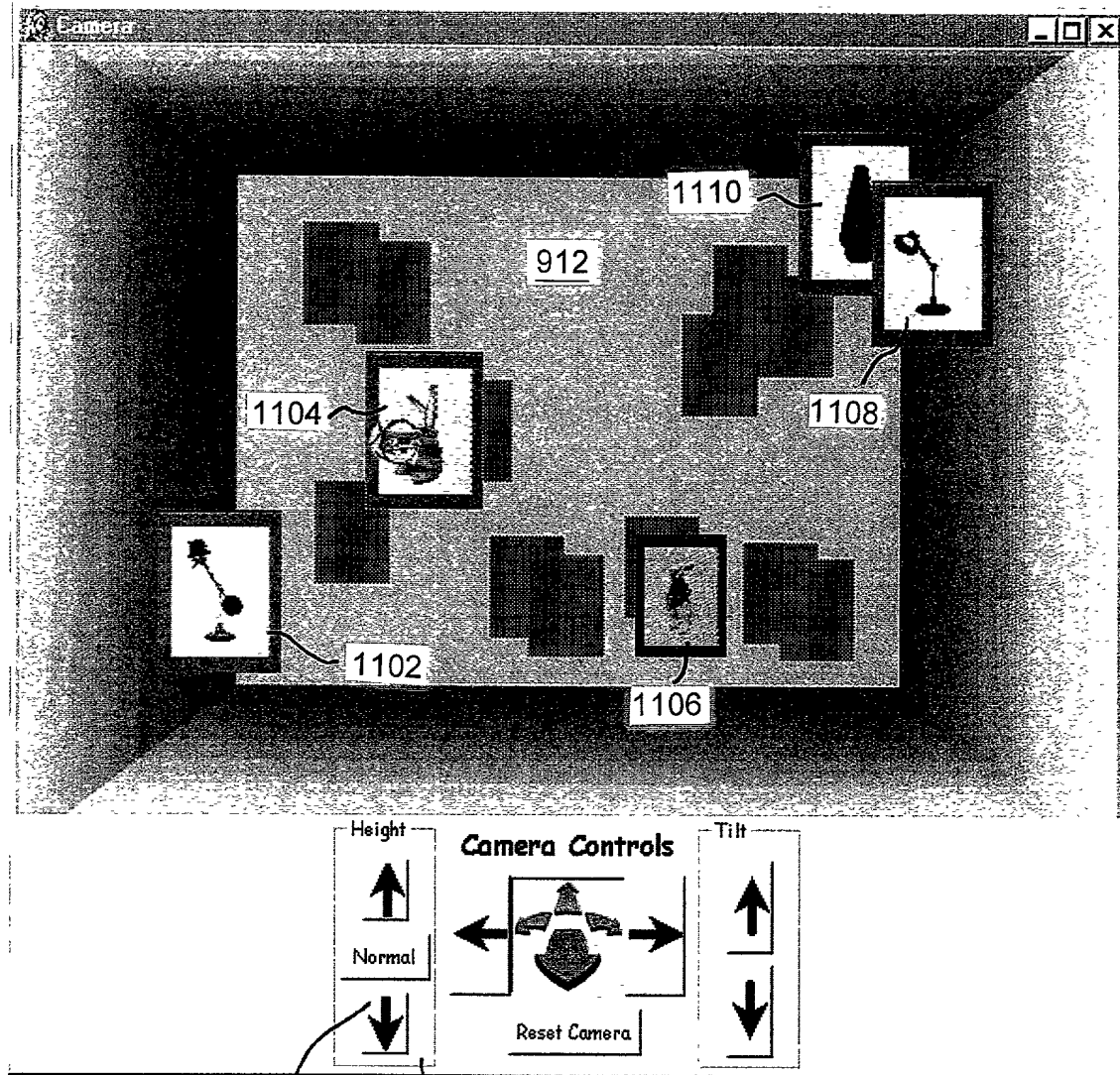


FIGURE 11A



931

930

1100'

**FIGURE 11B**

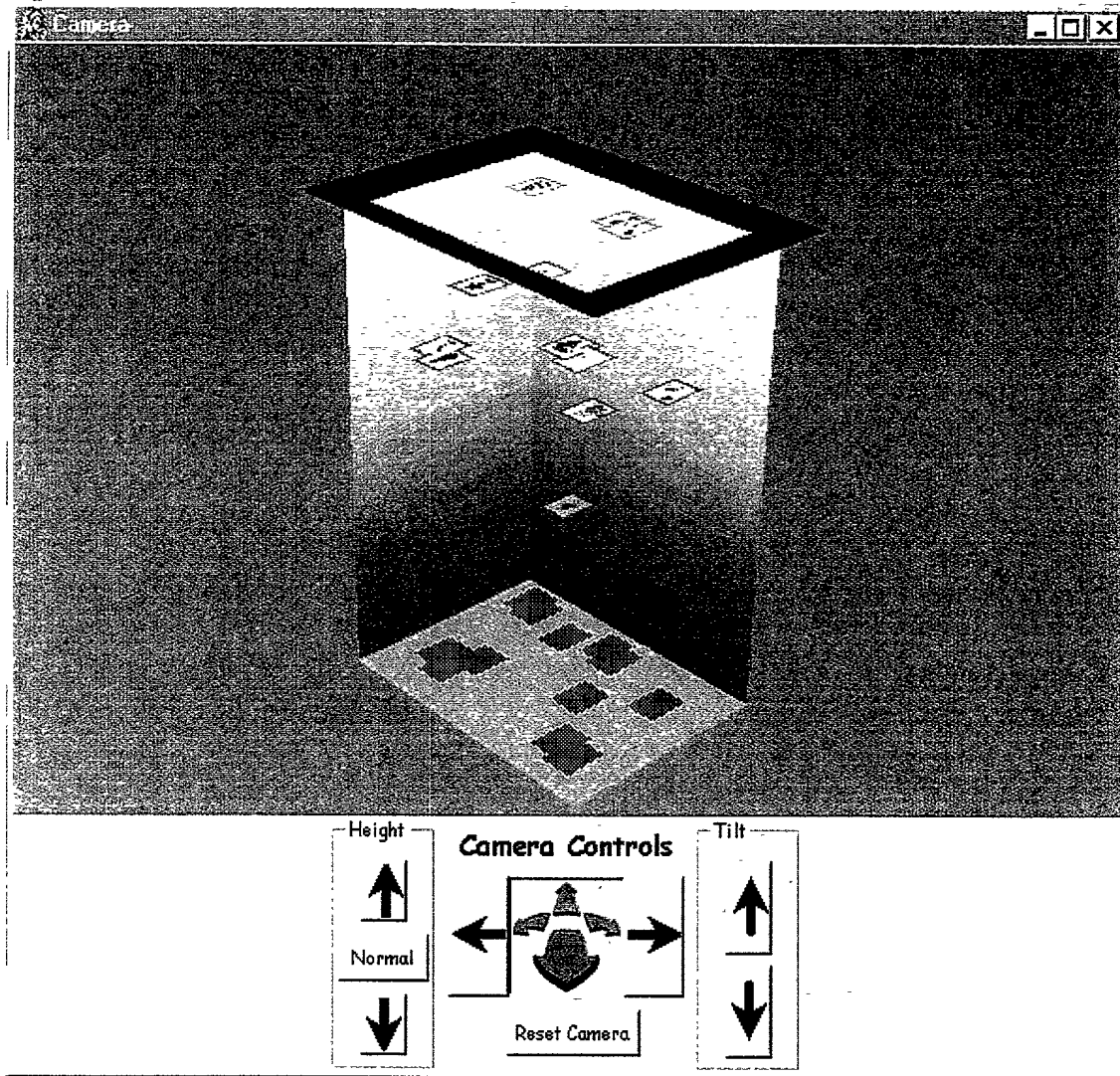
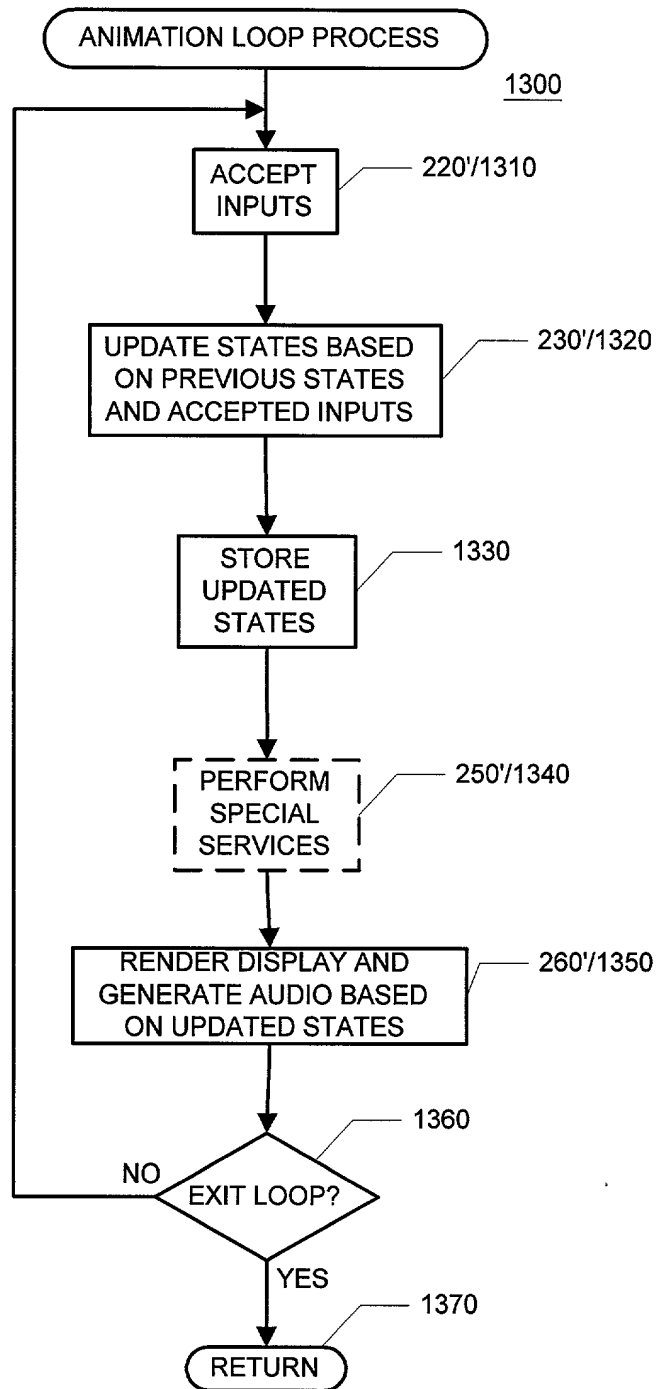
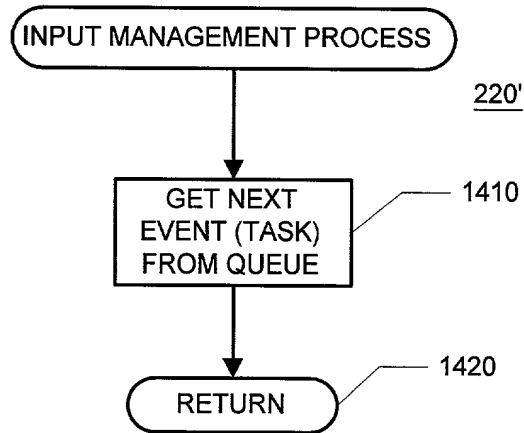


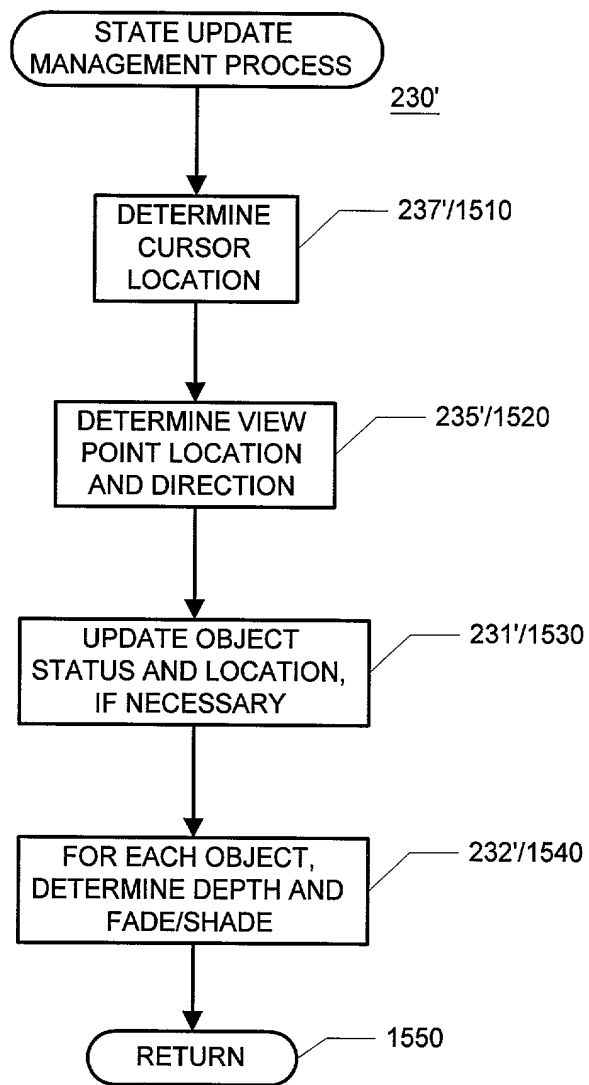
FIGURE 12



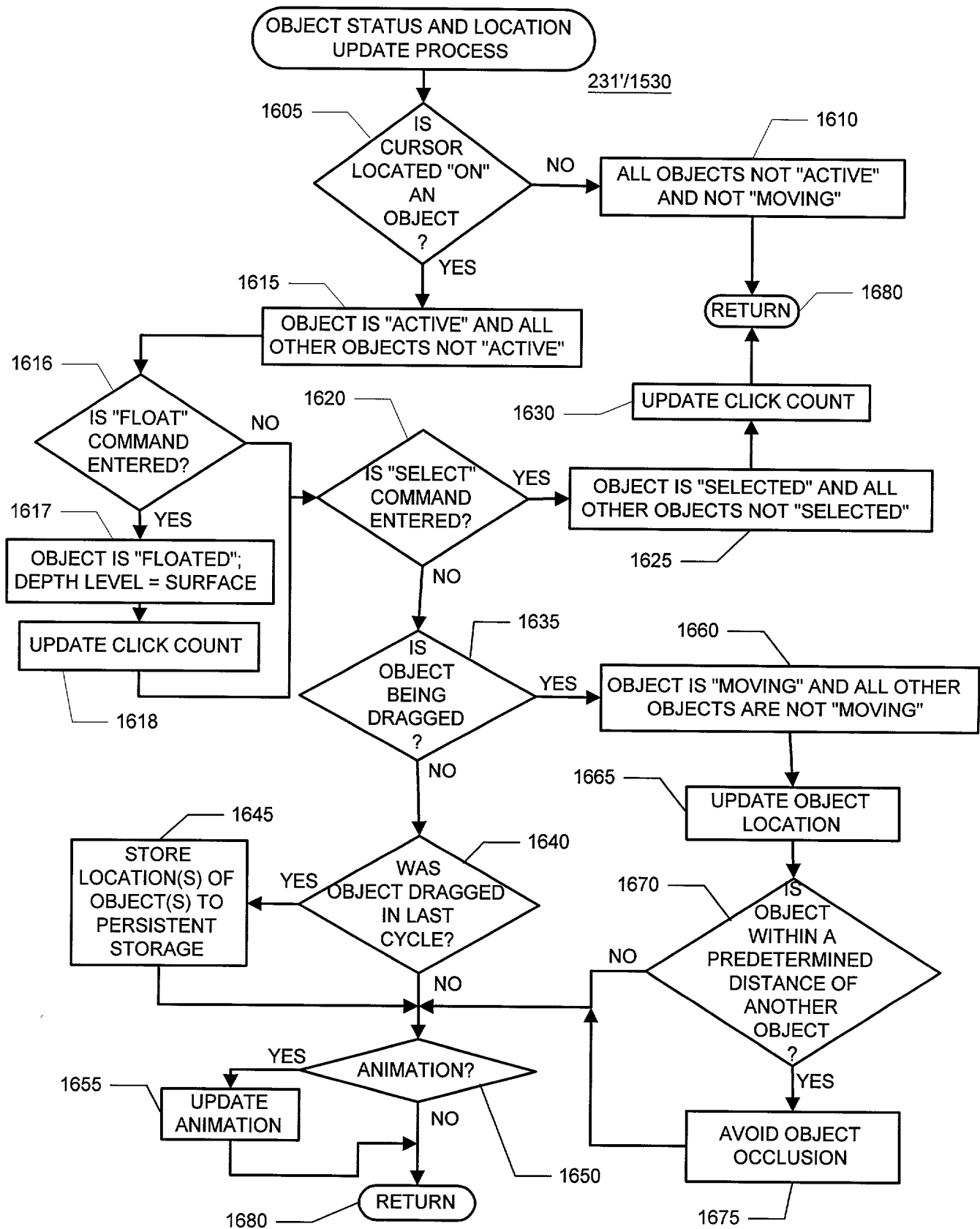
**FIGURE 13**



**FIGURE 14**

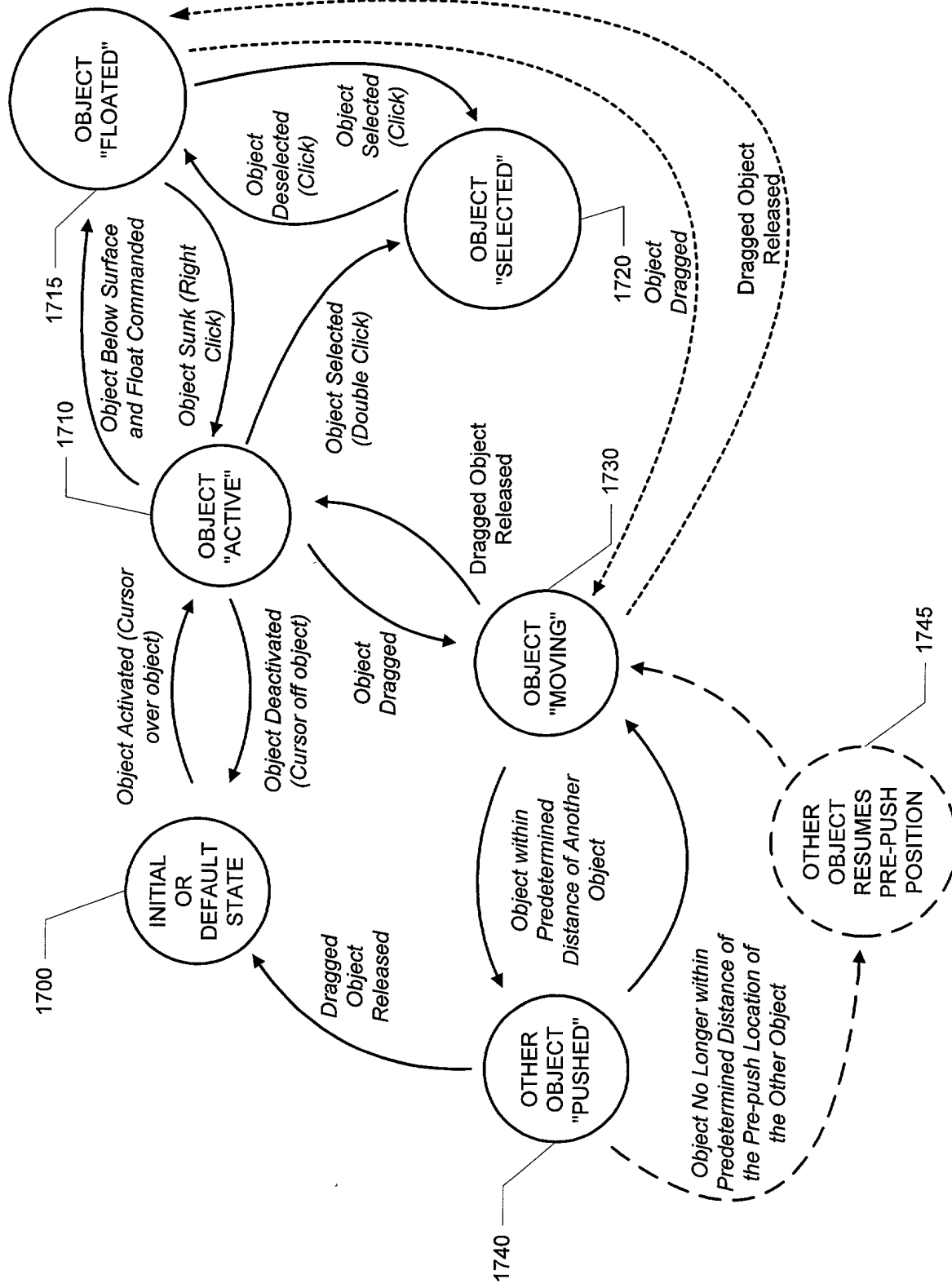


**FIGURE 15**

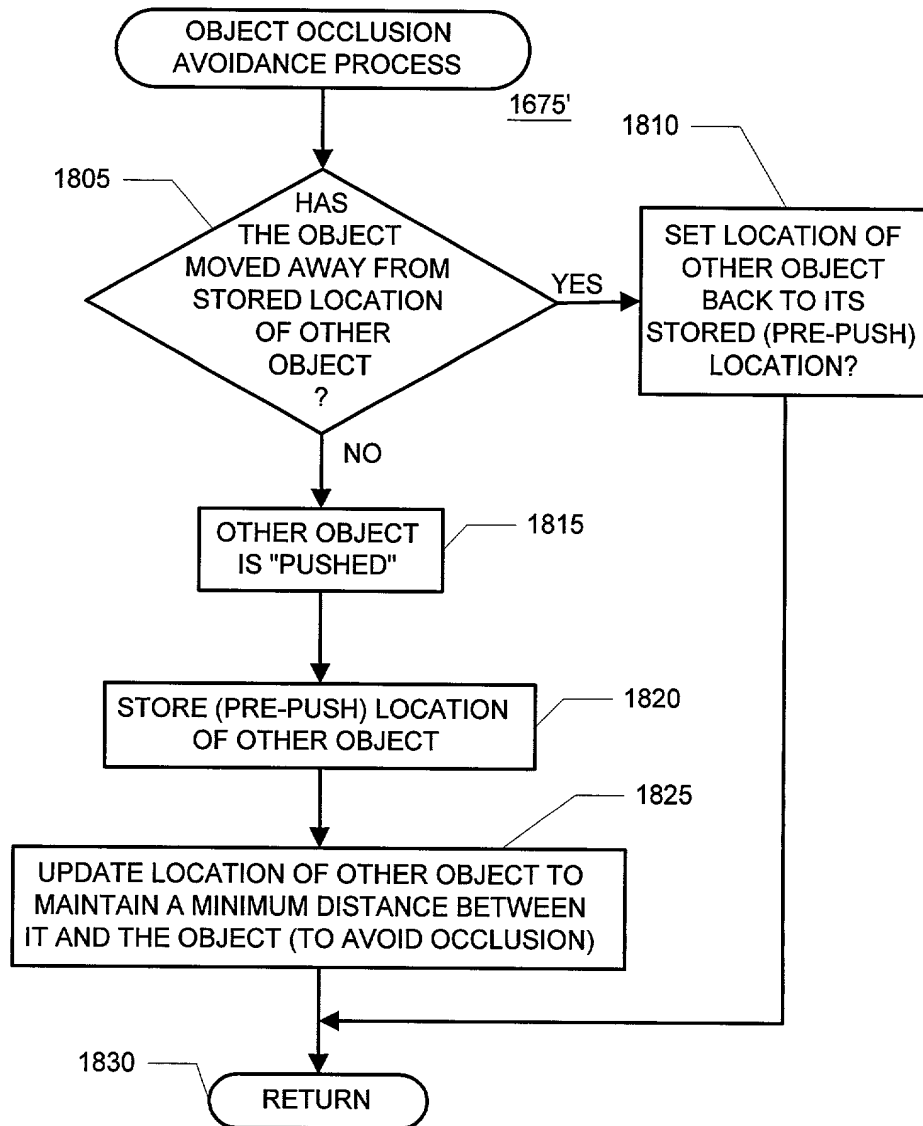


**FIGURE 16**

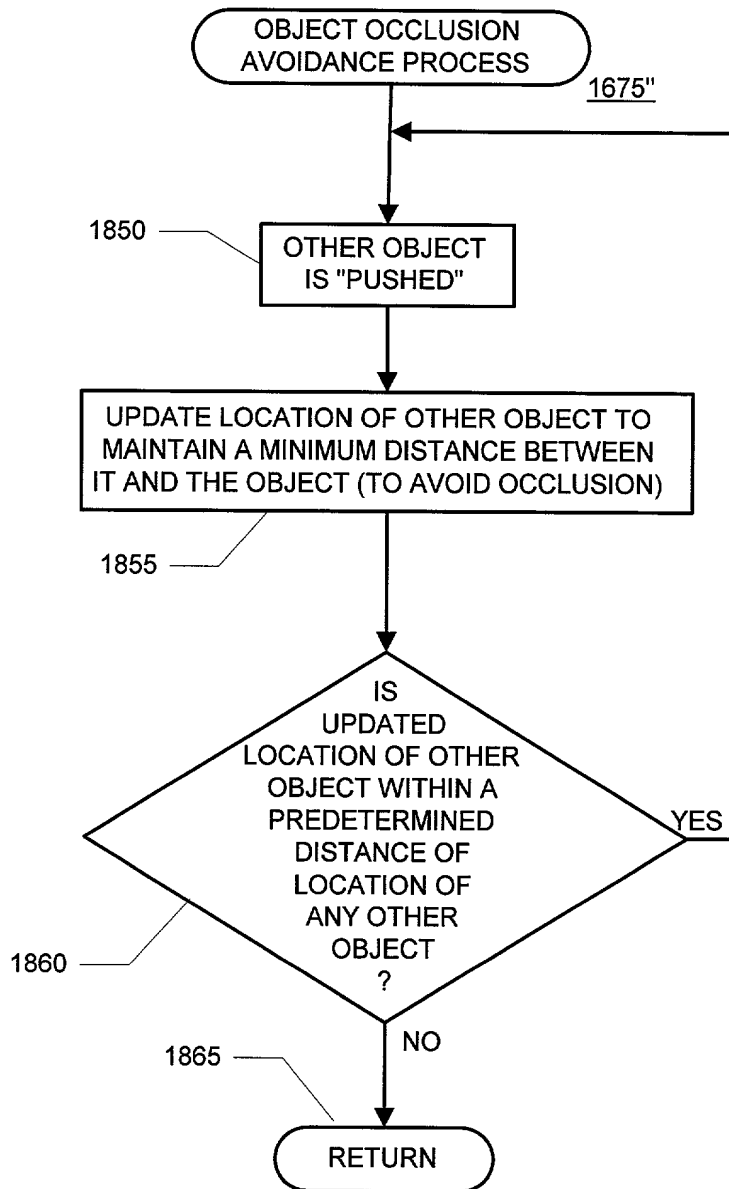
FIGURE 17







**FIGURE 18A**



**FIGURE 18B**

OBJECT DEPTH AND FADE/SHADE  
DETERMINATION PROCESS

232/1540

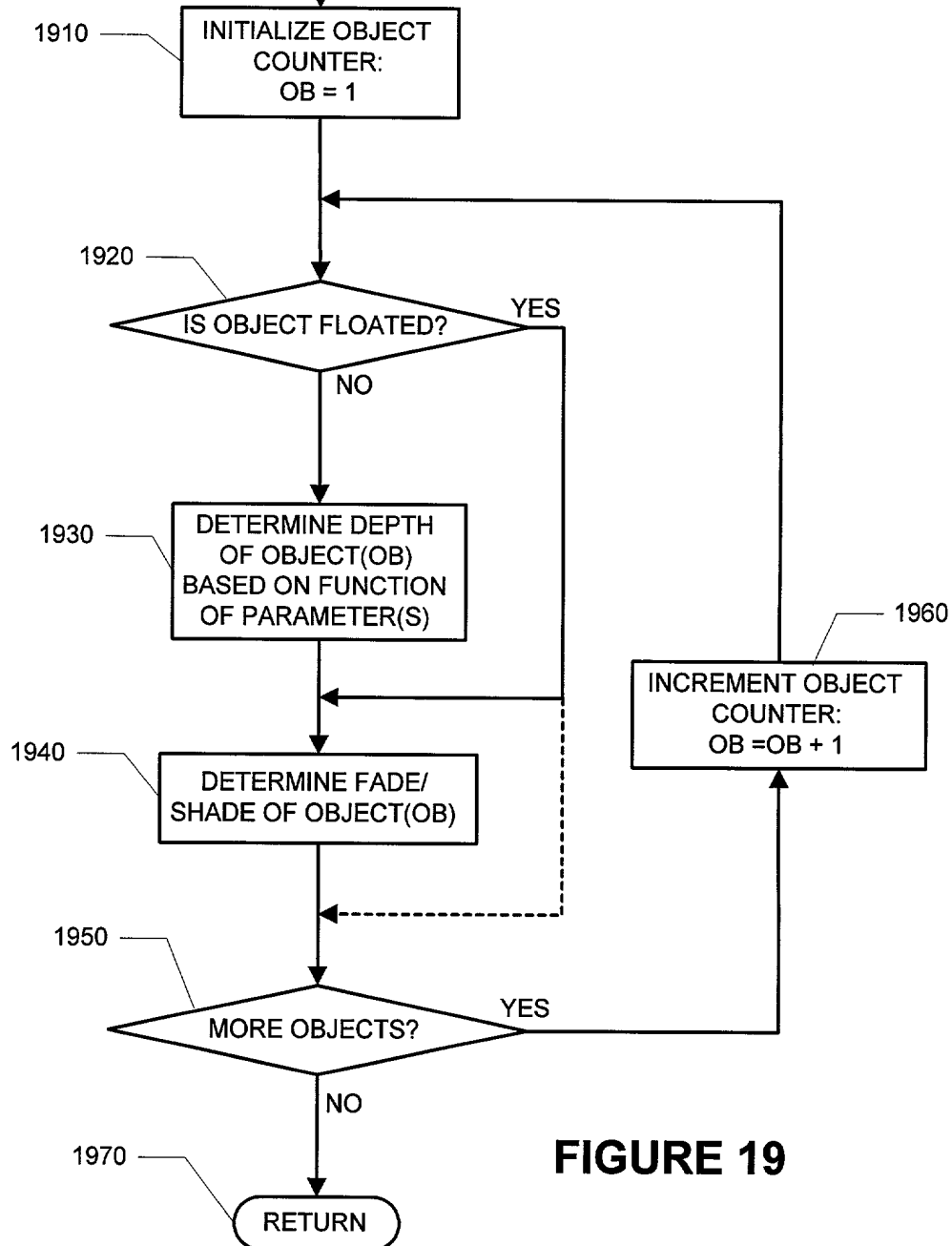
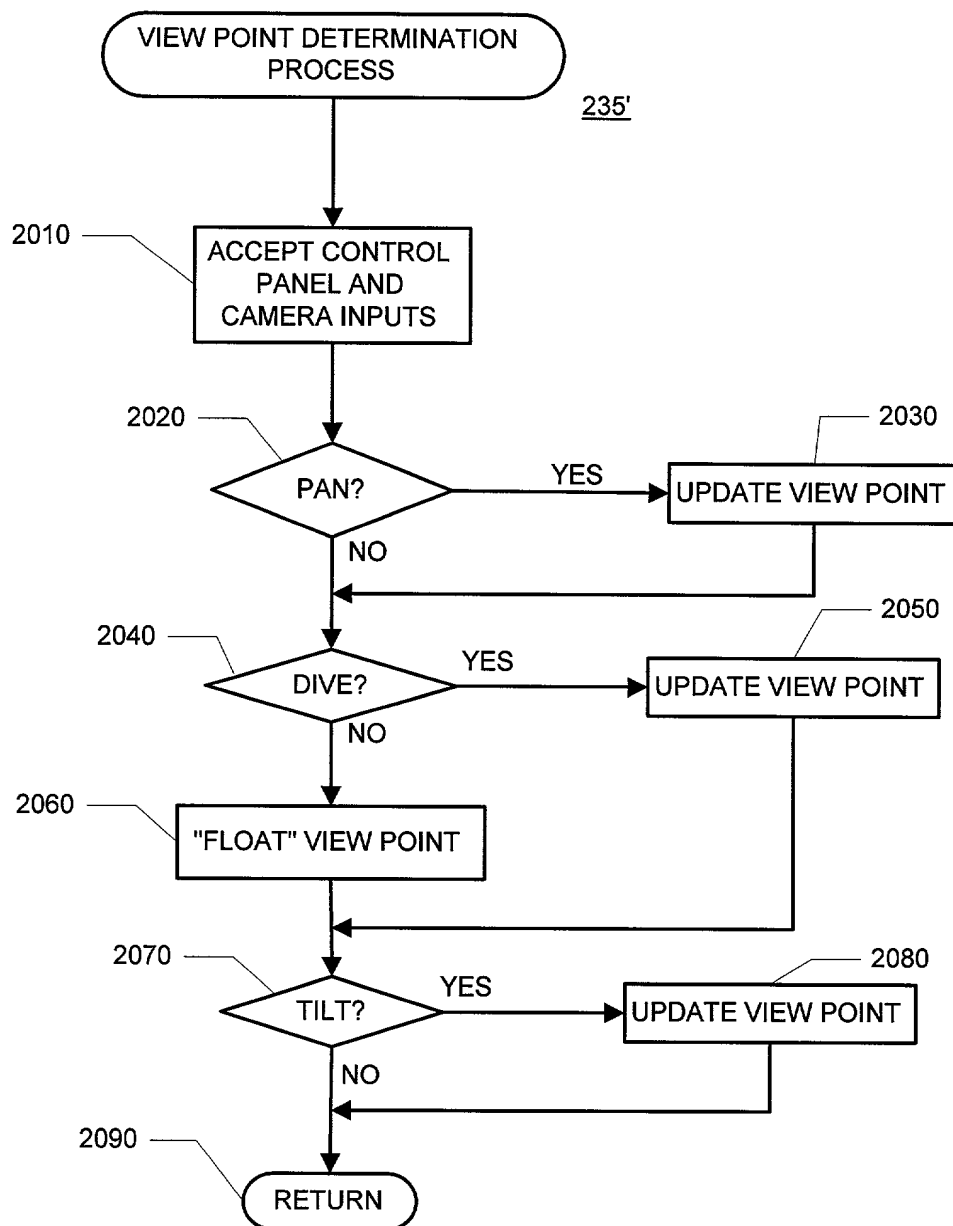


FIGURE 19



**FIGURE 20**

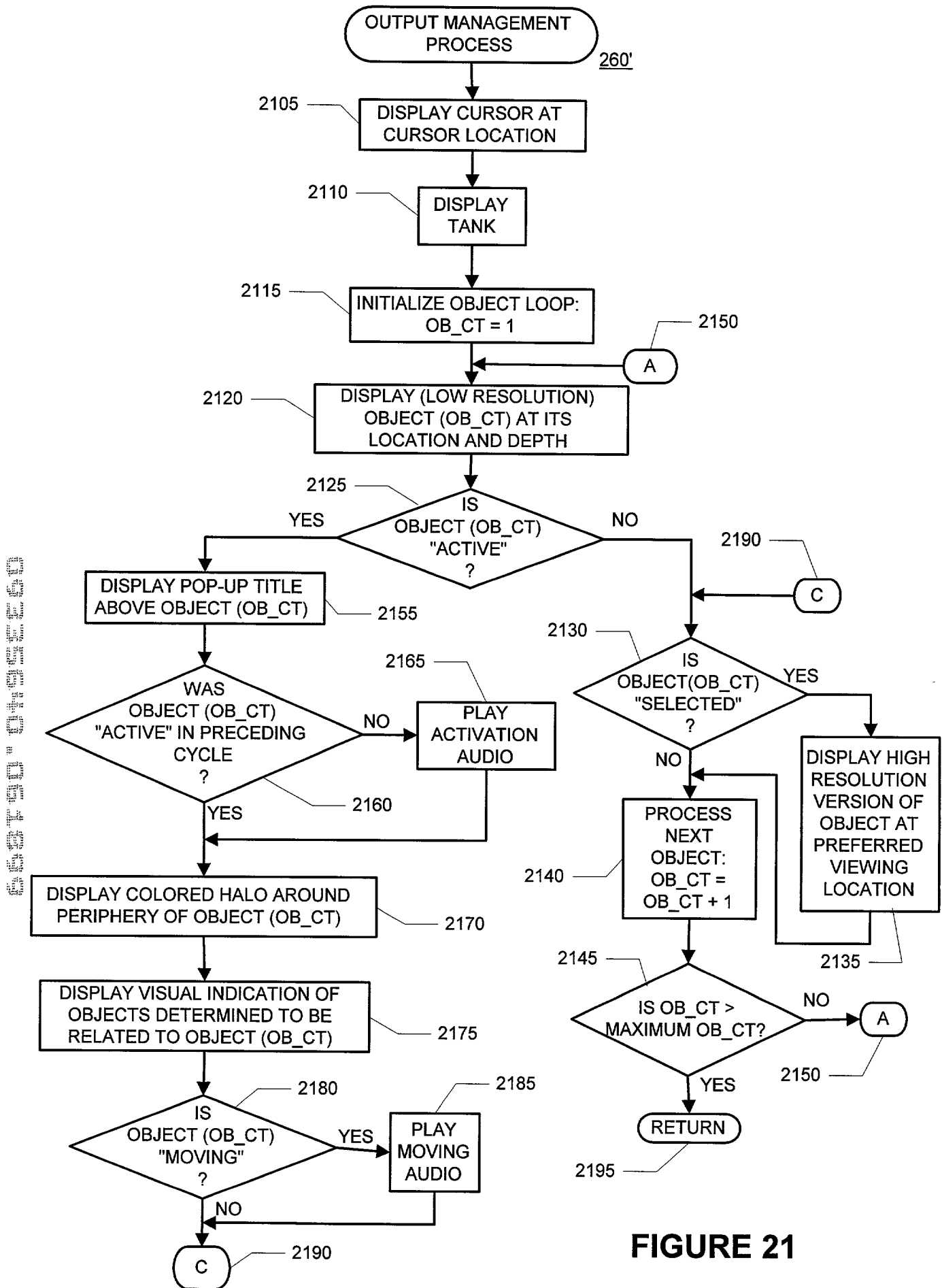
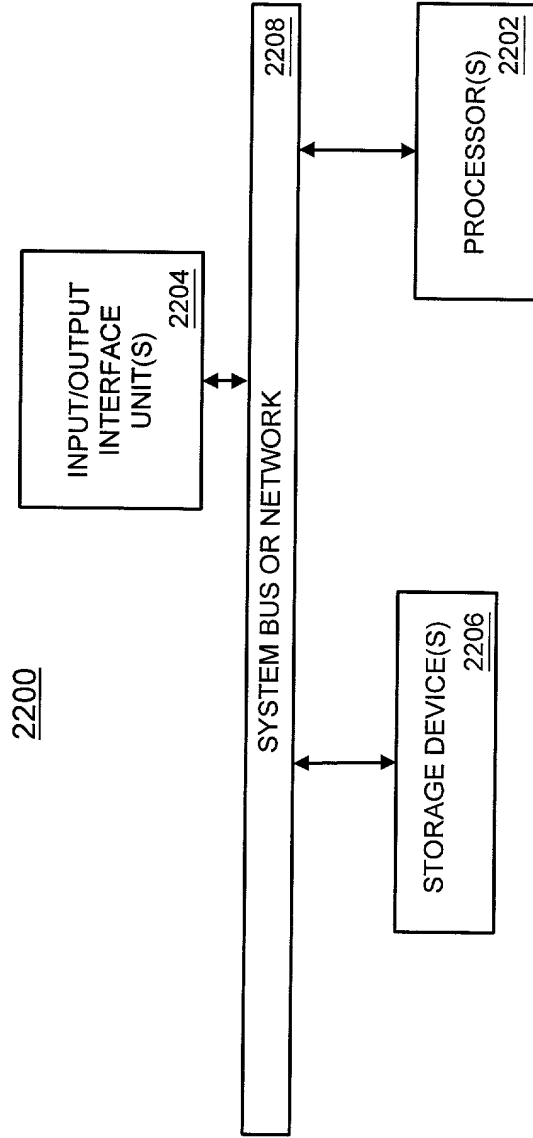


FIGURE 21



**FIGURE 22**